DETERMINATION OF OPTIMAL CUTTING PARAMETERS AND GROUPING OF OPERATIONS ON MULTI-SPINDLE AUTOMATICS

By
S. PADMANABHA

DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

JANUARY, 1979

DETERMINATION OF OPTIMAL CUTTING PARAMETERS AND GROUPING OF OPERATIONS ON MULTI-SPINDLE AUTOMATICS

A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY

By
S. PADMANABHA

to the
DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR
JANUARY, 1979

DEDICATED TO

MY PARENTS

AND

LORD VENKATESWARA

CERTIFICATE

Certified that this work on 'Determination of Optimal Cutting Parameters and Grouping of Operations on Multi-Spindle Automatics', by S. Padmanabha has been carried out under my supervision and that this has not been submitted elsewhere for a degree.

January, 1979

Dr. J.L. Batra
Professor
Dept. of Mechanical Engineering
Indian Institute of Technology
Kanpur 208016, INDIA

12.3.79

ACKNOWLEDGEMENTS

I take this opportunity to place on record my indebtedness to Dr. J.L. Batra for his valuable help and guidance.

My thanks are due to my friends H.K. Kanodia, R. Bajpai, Anay Srivastava and Akash Bahuguna for their timely help. Thanks are also due to Mr. J.K. Misra for his unquestionable typing and to Mr. Buddhi Ram Kandiyal for the careful cyclostyling.

S. Padmanabha

CONTENTS

LIST OF TABLES AND FIGURES

NOMENCLATURE

a_{mtn}	–	Exponent of the n-th variable of the t-th term in the m-th polynomial; m = 0 corresponds to the objective function.
C_{mt}	–	Coefficient of the t-th term of the m-th polynomial.
CT	–	Cycle Time, min.
d	–	Depth of cut, mm.
D	–	Machining diameter, mm.
f	–	Feed, mm/rev.
$f_{m\ max}$	–	Maximum feed rate on machine tool, mm/rev.
$f_{m\ min}$	–	Minimum feed rate on machine tool, mm/rev.
F	–	Cutting force, Kgf.
H	–	Horse power, HP.
H_{max}	–	Maximum horsepower available on machine tool, HP.
K_1	–	Constant in Taylor's tool-life equation
K_2	–	Constant in Force equation
L	–	Length of cut, mm.
MT	–	Total machining time, min.
N	–	Number of spindles (or groups)
O	–	Set of numbers indicating the constraints that are violated
P	–	Total number of operations
PT	–	Processing time, min.
Q	–	Set of numbers indicating the constraints that are loose

R	–	set of numbers indicating the constraints that are tight
S	–	Speed, rpm.
$S_{m\ max}$	–	Maximum feed available on machine tool, rpm.
$S_{m\ min}$	–	Minimum feed available on machine tool, rpm.
t_c	–	Tool-change time, min.
t_m	–	Machining time for an operation, min.
T	–	Tool-life, min.
T_m	–	Number of terms in m-th constraint
T_o	–	Number of terms in objective function
y_o	–	Sum of terms of objective function
y_m	–	Sum of terms of m-th constraint
γ	–	Surrogatex multiplier for the m-th constraint
μ_1, μ_2, μ_3	–	Exponents of d, S and T respectively in Taylor's tool-life equation
μ_4, μ_5	–	Exponents of d and f respectively in the Force equation.
η	–	efficiency of transmission of power

The following notations are for the i-th spindle (or group):

d_{ij}	–	depth of cut for the j-th axial operation, mm.
d_{ik}	–	depth of cut for the k-th radial operation, mm
f_{ai}	–	feed rate for axial operations, mm/rev.
f_{ri}	–	feed rate for radial operations, mm/rev.
H_{ij}	–	horsepower for the j-th axial operation, HP.
H_{ik}	–	horsepower for the k-th radial operation, HP.

L_{ij}	-	length of cut for j-th axial operation, mm.
L_{ik}	-	length of cut for k-th radial operation, mm.
q_{ai}	-	total number of axial operations
q_{ri}	-	total number of radial operations
S_i	-	speed, rpm.
t_{cij}	-	tool change time for j-th axial tool, min.
t_{cik}	-	tool change time for k-th radial tool, min.
t_{mij}	-	Machining time for j-th axial operation, mm.
t_{mik}	-	Machining time for k-th radial operation, min.
T_{ij}	-	Tool life of j-th axial tool, min.
T_{ik}	-	Tool life of k-th radial tool, min.

SYNOPSIS

S. PADMANABHA
M.Tech. (Mech.)
Indian Institute of Technology
Kanpur
January, 1979

DETERMINATION OF OPTIMAL CUTTING PARAMETERS AND GROUPING OF OPERATIONS ON MULTI-SPINDLE AUTOMATICS

A solution methodology is proposed to determine optimal cutting parameters on multi-spindle automatic lathes with minimization of cycle time as evaluation criterion. The difficulty in attacking this problem arises from the following reasons:

1. Information regarding grouping of operations is necessary for optimization of cutting parameters.
2. This information is not available initially.
3. Besides precedence (technological) constraints the information about cutting parameters have also to be used for grouping, because operations which form a group are forced to assume the same magnitude for some of the parameters depending on the constraints of the machine-tool.

This interdependence between grouping of operations and cutting parameters optimization is sought to be overcome using a 3 phase procedure.

1. Phase 1: Optimization of cutting parameters considering the operations to be machined sequentially. The output of this phase provides an input to the next phase.

2. Phase 2: Grouping of operations based on the cutting parameters generated in Phase 1 and the precedence constraints.

3. Phase 3: Optimization of cutting parameters taking into account the constraints generated by the grouping of operations in phase 2 with minimization of cycle time as the objective function.

Optimization in phases 1 and 3 is carried out using Surrogate Geometric Programming, a relatively new technique, which takes advantage of the posynomial formulation of the problems. In phase 2 a quantitative approach based on Cluster Analysis is used for the grouping of operations.

The methodology has been implemented on IBM 7044. For a sample problem given by Bartalucci et al using the same input data for the test problem, phase 1 yielded

20 variables and 41 constraints while phase 3 involved 9 variables and 29 constraints for a 3 spindle automatic lathe. It was observed that the proposed approach yields speedier results as compared with the results reported by Bartalucci et al in terms of lower production cycle time and higher computation efficiency.

CHAPTER I

INTRODUCTION

1.1 THE PROBLEM - AN OVERVIEW:

The constantly increasing costs of labour, machine tools and materials and growing competitive activity, both domestic and foreign, are placing new challenges before the metal cutting industry. Metal forming by the chip producing methods must be done, not only faster, but at a lower unit cost, if profit is to be realized. Inefficiencies in machining practices had been and is being tolerated or disregarded by many for years, because no real squeeze has been felt. The era of such complacency has to end for most, and the time will rapidly run out for the few remaining.

This thesis is concerned with the problem of determining optimal machining conditions on multi-spindle automatic lathes. Given a job to be machined on a multi-spindle automatic, it is necessary to determine the optimal machining parameters of speed and feed for each of the required operations and to assign the operations to the available spindles, the latter being referred to as grouping of operations. For better understanding of the problem, the important characteristics of multi-spindle automatic lathe are described next.

A horizontal multispindle automatic lathe has a number of stations (or spindles) available for cutting operations; on each of them one axial and one radial carriage are available. Cutting tools are mounted on the carriages according to the machining requirements and the grouping of the operations on the spindles.

The workpiece is progressively machined at each station and is indexed to the next station only after machining is finished at each station and the carriages carrying the cutting tools have been withdrawn from the workpiece. One station is reserved for loading and unloading which is performed either manually or mechanically.

As far as the spindle motor is concerned, the following schemes are in vogue.

1. Single power unit driving all the spindles at the same speed
2. Single power unit driving all the spindles at different speeds
3. Separate power unit for each spindle.

The most commonly used are schemes 1 and 2, while scheme 3 is used in special machines and in transfer lines.

Coming back to the problem at hand we notice that for the optimisation of cutting parameters of the various operations one needs information regarding the grouping of

operations. However, the grouping of operations is to be based not only on the precedence requirements of the various operations but also the values of the cutting parameters selected for various operations. This is due to the fact that while constituting a group of operations for a station (spindle) the following important aspects must be borne in mind. As far as possible,

1. the group comprises of operations having the same optimal speed (rpm)
2. for operations to be performed by the axial slide, the optimal feed rates should be the same for all the operations
3. for operations to be performed by the radial slide, the optimal feed rates should be the same for all operations.

In order to overcome this difficulty which arises due to interdependence of the cutting parameters and the grouping of operations, a three phase procedure has been evolved.

In phase 1 the problem is formulated assuming that all the operations of the given job are machined sequentially and the evaluation criterion is optimised. The output of this optimisation, i.e., the optimal speeds and the feeds for individual operations, are then exploited for the formation of groups in phase 2.

In phase 2 the output of phase 1 is used to form feature vectors, which describe the characteristics of each operation. These feature vectors are then grouped using a quantitative technique called Cluster Analysis. The output of this phase then is a grouping of operations which provides the input for phase 3.

In phase 3 the final optimisation is made after reformulating the problem so as to account for the grouping of operations. The output of this phase will provide the optimal cutting parameters.

Surrogate Geometric Programming technique is used for optimisation in both phase 1 and phase 3 to take advantage of the posynomial structure of the formulation.

1.2 OVERVIEW OF THESIS:

The main body of the thesis is concentrated in Chapter III and literature survey is given in Chapter II.

In Chapter III, problem formulation and solution methodology are presented. A relatively recent optimisation technique, Surrogate Geometric Programming is presented as a solution methodology for optimisation in phase 1 and phase 3. The core of this algorithm is a geometric program suggested by Blau [5], while the scheme for updating the surrogate constants was developed in the course of this work. The solution methodology for grouping of operations

using a quantitative technique called cluster analysis is also presented. The two solution methodologies were computerised and implemented on IBM 7044.

The proposed methodology was tested on a test case used by Bartalucci et al.[3]. The same set of input parameters were used. The test case requires ten operations. Therefore, phase 1 formulation involves twenty variables and fortyone constraints. The results of the application of the three phase procedure are discussed in Chapter IV.

CHAPTER II

LITERATURE SURVEY

In this work a brief review of the work reported in the literature regarding cutting variables optimization and grouping of operations (on multi-spindle automatics) is presented. Since tool replacement policies are closely linked with cutting variable optimization, some of important contributions in this area are also surveyed.

2.1 OPTIMIZATION

2.1.1 Optimization of Cutting Variables:

The problem of selection of optimal machining conditions in the basic turning operations has been analysed until recently with varying degrees of generality by many investigators. Gilbert [28], Brewer and Rueda [8] studied the economics of machining and considered the influence of cutting speed, whereas, other authors, Brewer [7], Brown [9], Brewer and Rueda [8], Cook [10], analysed the effect of both cutting speed and feed rate. The afore-mentioned analyses were based on the minimization of unit cost. Unit cost and production rate were also optimized, by Field et al [24], by determination of the optimal value of cutting speed for turning, milling, drilling, reaming and tapping.

Brewer [7], Radford and Richardson [59] considered the problem of optimizing unit cost in turning taking into account the cutting power available as a constraint and Petropoulos [54] studied the constrained problem of the selection of optimal machining conditions in face milling by using unit cost and production rate as optimal criteria.

Goranskii [30], in a rigorous mathematical analysis expressed the limitations imposed by machine tool, workpiece and cutting tool as power transforms, solving the relationships by linear programming.

The first use of computers in optimization seems to have been reported by Doris et al [15] in which a computer program was used to evaluate all combinations of speed, feed and depth of cut, seeking the minimum overall cost. The limitations imposed by tool wear, power, and workpiece rigidity were also taken into account.

Brown [9] developed an analytical method for studying the effect of various parameters in turning in two passes.

Brewer [7] suggested the use of Lagrange multipliers for the optimization of the constrained problem of unit cost, with mainly cutting power as constraint, but did not proceed with the analytical optimization procedure.

Bhattacharya et al [4] optimized the unit cost for turning subject to constraints of surface roughness and cutting power by the use of Lagrange's method of coupled extremes. This method yielded a system of highly nonlinear equations the solution of which was effected through a computer.

Optimization in practical workshop conditions was considered by Ham and Gonzalez [33] by iteratively updating the tool life equation from actual workshop production results till convergence is obtained and then the optimal conditions are determined.

Garina et al [27] proposed experimental determination of optimal cutting conditions with the help of Simplex Method (Nelder and Mead).

Cost optimal speed for N.C. Lathes was determined by Vulfson and Deryabin [74] by first considering the feed to be constant and then cost optimal feed was found by an exhaustive calculation over all possible feed values. Gurevich [38] later modified this algorithm to take into account the Taylorized range.

Reznikov [68] optimized time for a single operation with speed and feed as variables using linear programming. The constraints considered were speed, feed, horse power, surface finish and quality of sub-surface layer.

Kops [43] studied the effect of constant cutting speed and constant rpm on the cost and time parameters when machining stepped parts with large and small variation in diameters, while Crookall and Venkataramani [11] optimized speed, feed and depth of cut with respect to time for multi-pass turning with power, deflection, surface finish and tool wear constants.

To bridge the gap between theoretically optimal and practically optimal values two approaches are available, namely, determination of correction factors experimentally and incorporation of the probabilistic nature of tool life and cutting force into the optimization procedure.

Thus Kolev [42] suggests a graphical method to determine the speed and feed by means of a cyclogram constructed for a particular tool and workpiece material combination, and Romanov and Sotnikova [69] present correction factors obtained after extensive experimentation considering factors such as set-up, part-clamping, etc.

Iwata et al [36] proposed an analytical method applying a chance constrained approach to determine cost optimal cutting conditions - speed and feed - considering the probabilistic nature of objective function (tool life exponents) and constraints (cutting force exponents). A solved example indicated a rather insignificant change

- 2.594 yen/piece (1 paise/piece) - for a change in probability level of tool life from 0.5 to 0.99.

In a more recent work Iwata et al [37] optimize the speed, feed and depth of cut with respect to cost. Depth of cut is first optimized using dynamic programming after which a chance constrained approach and SUMT are used to optimize the speed and feed as in the previous [36] work. Here again one notices an insignificant change, that too only in the feed, of 0.0033 mm/rev for a variation in probability level of the constraints from 0.5 to 0.99. This is insignificant in the sense that unless the machine is meant to be of high precision, such precision is not normally available. Effect of tool life (objective function) probability level variations were not presented.

Geometric Programming (GP) has also been used by many investigators for the determination of optimal cutting parameters. Since the present work also uses G P, an extensive review for the same is presented in the following section.

2.1.2 Geometric Programming

2.1.2.1 The GP Technique:

The first extensive work on Geometric Programming was brought out by Duffin, Peterson and Zener [17]. In the latest work on the subject Phillips and Beightler [58]

covered all the versions of Geometric Programming available till 1975 and discussed the merits and demerits of each along with a large number of engineering and other applications.

Ragsdell and Phillips [60] used the condensation version of G P to design a class of welded structures, which problem consisted of 9 nonlinear constraints, 7 variables and 16 DOD.+ The results obtained are compared with those obtained using 8 different, conventional optimization techniques.

Mancini and Piziali [48] developed a strategy based on G.P. using normality and orthogonality conditions of the dual problem to define candidate vertices and then investigating the solution at each vertex for the purpose of optimally designing helical springs.

Dinkel and Kochenberger [14] present a post-optimal sensitivity analysis procedure using G P as the basis, in which the effect of changes in the coefficients are related to the design variables.

Hwner and Mehta [35] use the Blau algorithm for G.P. to optimize computer performance with throughput and utilization as criteria. The problem formulated had 2 variables, one constraint and 7 DOD.

+ Footnote on page 12

Nicoletti and Mariani [50] also used the Blau algorithm to solve two separate problems in management-structural control in a graded manpower system and advertising scheduling - the latter consisting of 5 constraints, 9 variables and 15 degrees of difficulty (DOD)+.

Rao et al [64] criticized the Blau algorithm - because it involves a matrix inversion of a high order at every stage of iteration - and the algorithm developed by Templeman et al because one has to resort towards the end to crawling to obtain satisfactory convergence - and presented a new hybrid algorithm based on Liapunov concepts.

2.1.2.2 G.P. and Machining Variables Optimization:

Phillips and Beightler [57] first applied G.P. to optimize unit cost with speed and feed as variables and one, horse power, constraint. The maximum degree of difficulty (DOD) involved was 1. Walvekar and Lambert [75] solved a similar problem but with two constraints of horse power and surface finish using G.P. Here also the maximum DOD was 1.

+ with respect to the primal of G.P. formulation the difference between the number of variables (N) plus unity, and the number of terms (T), including the constraint terms, is called degrees of difficulty. It indicates the difficulty involved in solving the dual of the G.P. program where there are a system of (N+1) linear equations in (T) variables.

Ermer [20] solves the machining cost optimization problem with G.P., with constraints of feed, surface finish and horse power. For the problems solved, the maximum DOD involved was 2.

Petropoulos [54] compares G.P. with conventional methods in optimizing the unit cost for the turning problem. The numerical examples involve a DOD of 1.

2.1.3 Multi-Spindle Automatics:

The first optimization effort in this area seems to be that of Gilman et al [29] where the optimization technique used is not available but essentially consisted of first selecting the feeds from reference data. The criteria considered were machining time, operation cost and tooling expenditure.

Bartalucci et al [3] optimized the cutting conditions on multi-spindle automatics using a method, which borders on exhaustive enumeration, whereby the constrained range of variation of each variable is first fixed, after which, starting from the maximum value, it is progressively decreased until the rate of change of cost becomes zero.

Kapustin et al [39] considered the case of vertical multi-spindle automatics. After grouping the operations onto the spindles the optimal speed and feed for each group

were determined from the available discrete speeds and feeds by complete enumeration. Knowing the group which requires the maximum cutting time, the speed-feed combinations for the other groups were adjusted to equalise this maximum time, as far as possible, using an assignment algorithm with minimization of torque as criterion. It may be noted here that it is not necessary that discrete feeds be assumed, which may be inadvisable in mass production environment where cams controlling feeds can be made to order.

Kothari [44] attempted probabilistic optimization of cutting variables on multi-spindle automatics but incorrect formulation of constraints invalidated both the approach and the optimization results.

2.1.4 Tool Replacement:

Optimal tool replacement policies are closely inter-linked with optimal cutting parameters since it is the latter which control the tool-lives, which in turn influence the optimal tool replacement policy. In this section a brief survey of tool-life models and tool replacement strategies is presented.

2.1.4.1 Tool-Life Models:

The use of probabilistic tool-life models is a relatively recent phenomenon the latest of which, in available literature, are that of Ramalingam and Watson [63]

who developed a 'single injury tool life model' purely on the theoretical basis of the relevant physical wear mechanism involved. Ramalingam [62], in continuation of the previous work, developed models for 'multiple-injury tool-life', again on a purely theoretical basis, but in a discussion Cohen and Black [62] validate Ramalingam's model by comparison with eleven experimental test reports on tool life.

A rather interesting model of the mean life of equipment subject to compound failure was developed theoretically by Malik [46] who draws the attention of researchers toward developing a relevant integral transform of this function.

2.1.4.2 Tool Replacement Strategies:

Kothari [44] mentions four reported attempts at obtaining an optimal tool replacement policy. The first, suggested by Taha, is a cost model for a strict interval - replacement policy where the cost of rejects is taken into account. In the second, Ebert and Hershaver obtained an optimal with respect to tool - operating costs through simulation, the parameters were quality control decision rules and strict-interval replacement. Strict interval replacement was reportedly found better. In the third, Okushima and Fujii formulated cost models for scheduled and

unscheduled tool replacement policies and concluded, after simulation, that scheduled changes are more economical. The fourth by Duncan is connected with mathematical and numerical models for scheduling cutting-tool changes.

Esterzon and Radzievskii [21] introduced the concept of optimal group-tool changes for which the stoppages in transfer lines were a minimum. This work is based on time studies of the tools in a transfer line.

Sheparov [71] suggested a simple graphical method for planning optimal group tool changes which consisted essentially of grouping the tools with similar tool-life magnitudes into a predetermined number of groups.

Pas' Ko [51,52,53] in three separate works, developed mathematical models to determine both individual and group optimal tool replacement policies. A Weibul tool-life distribution was assumed.

2.1.4.3 Related Topics:

A tool edge can be used for a number of different operations depending on its life. So, when machining a number of pieces consequentively on automatic machines the necessity of optimal tool selection arises. This topic has been dealt with by Etin et al [22].

De Vor et al [12] conducted experiments to determine the nature of tool-life variation as a function of pre-specified wear level and introduced the concept of optimum wear level specification as a result.

2.2 GROUPING OF OPERATIONS:

Once optimization is completed it becomes necessary to group the operations such that they are distributed among the available spindles according to some accepted criteria.

2.2.1 Multi-Spindle Automatics:

Only three cases of attempts at such a grouping for multi-spindle automatics were reported in available literature.

Thus Jones and Morgan [38] used the AIDA (Analysis of Interconnected Decision Areas) technique to eliminate the technologically impossible combinations of operations and then examined all the remaining alternatives to determine the cost optimum combination, while Bartalucci et al [3] used a computer program to enumerate all the combinations, eliminate the technologically infeasible ones and scan the rest for the cost optimal one.

Kapustin et al [39] used a 6 digit binary code to progressively group operations, the only criteria for grouping being that the combination should be technologically feasible.

2.2.2 Cluster Analysis:

This technique of grouping 'objects' which are 'similar' with respect to some criteria appears appealing as it may be possible to suitably incorporate the grouping optimization criterion into the feature vector of each 'object'.

Duran and Odell [18] provided an excellent survey of different clustering approaches and distance functions, while Hartigan [34] and Anderberg [1] made available to prospective users different versions of each technique along with their computer programs.

Duda and Hart [16] discussed the use of various clustering algorithms with respect to pattern classification.

An important report of computational experience was given by Fromm and Northouse [26] who also presented empirical relations for lumping and splitting of clusters.

A review of cluster analysis is given by Kennedy in [41].

The next chapter presents the problem formulation and solution methodology for the problem of optimization of cutting variables and grouping of operation on multi-spindle automatics.

CHAPTER III

PROBLEM FORMULATION AND SOLUTION METHODOLOGY

3.1 INTRODUCTION

The type of automatic lathe considered in this thesis is as follows

i) A single motor drives all the spindles. Continuously varying speeds are available on all the spindles within a specified range.

ii) Different continuously varying feeds are available for axial and radial slides on each spindle. Further, the range of feed variation for both types of slides are equal.

The minimization of cycle time is considered as the criteria of optimization.

Given a job to be machined on an automat, it is necessary to know the grouping of operations onto the available spindles before we determine the optimal cutting parameters for the various operations to be performed on the automat. However, grouping of operations cannot be done without knowledge of optimal cutting parameters for the various operations. This being necessitated by the

fact that after grouping, in each group, all operations have a single speed, all radial operations have a single feed and all axial operations have a single feed. The difficulty caused by this interaction between grouping of operations and cutting parameters is overcome through the use of a 3 phase procedure. An overview of the 3 phase procedure is presented next. The complete details are discussed in Sections 3.2, 3.3 and 3.4.

Phase 1: In this phase the problem is formulated assuming that the operations required for the job are executed sequentially and the optimization criterion adopted is minimization of total machining time. The following constraints are considered:

i) maximum and minimum feed limits for each operation according to the cutting force, surface finish and machine tool,

ii) maximum and minimum speed limits for each operation according to the taylorized range and surface finish,

iii) maximum and minimum spindle speeds of the machine tool, and

iv) total power available on the multi-spindle automatic. The total power requirement for all operations should be less than or equal to the power available.

As would be explained in Section 3.2, the problem is posynomial in character and therefore Surrogate Geometric Programming (SGP) has been used for solving the optimization problem.

Phase 2 : The optimal cutting parameters obtained in Phase 1 are used to construct a feature vector which describes the characteristics of each of the operations and cluster analysis is used to find a grouping of operations taking into account the precedence constraints. The output of this phase yields grouping of operations.

Phase 3: The output of phase 2 provides the grouping of operations required for the final optimization which is carried out in this phase. The problem is formulated considering the operations to be machined in groups. The optimization is carried out considering the various groups of operations such that the cycle time is minimized. It needs to be pointed out that for the multi-spindle automatic lathe under consideration, the various groups of operations are machined simultaneously. Of the various operations, the operation requiring the longest processing time (PT) dictates the cycle time (CT) for the job. The processing time

is defined as sum of the machining time and the average tool change time per job, for the tool concerned.

The following constraints are considered:

1) All the operations in a particular group must have ~~xh~~ the same speed. Further, the selected speed should be within the specified maximum and minimum speed limits of the machine tool.

2) The feed for all radial operations of a group should be equal subject to maximum and minimum limits for feed.

3) The feed for all axial operations of a group should be equal subject to maximum and minimum limits for feed.

4) The total power requirement should be less than or equal to the total power available on the machine tool.

5) For no operation the processing time should be more than the cycle time (CT).

3.2 PHASE 1: OPTIMIZATION BEFORE GROUPING:

The problem for this phase is formulated assuming the operations are machined sequentially subject to the various constraints.

3.2.1 Objective Function:

The evaluation criterion is the minimization of total machining time which is given by

$$MT = \sum_{i=1}^{P} \left(t_{mi} + t_{ci} \frac{t_{mi}}{T_i}\right) \qquad (3.1)$$

where t_{mi} = machining time for i-th operation (min.)

$= \frac{L_i}{S_i f_i}$

L_i = Length of i-th operation (mm)

S_i = cutting speed for the i-th operation (rpm)

f_i = feed for the i-th operation (mm/rev.)

P = total number of operations

t_{ci} = tool change time for i-th tool (min.)

T_i = life of the i-th tool (min.)

The tool life T_i is obtained from the following expression [3]

$$S_i \, d_i^{\mu_1} \, f_i^{\mu_2} \, T_i^{\mu_3} = K_1$$

where K_1 = a constant,

d_i = depth of cut for i-th operation,

μ_1, μ_2, μ_3 are exponents for d_i, f_i and T_i respectively.

If eqn. (3.1) is expressed in terms of speed and feed we have

$$MT = \sum_{i=1}^{P} \left(C_{1i} \, S_i^{a} \, f_i^{b} + C_{2i} \, S_i^{g} \, f_i^{h}\right) \qquad (3.2)$$

where $C_{1i} = L_i$

$$C_{2i} = (L_i \, d_i^{\mu_1/\mu_2} / K_1^{1/\mu_3})$$

a, b, g and h are exponents given by

$$a = -1; \quad b = -1 \; ; \quad g = \frac{1}{\mu_3} - 1 \; ; \quad h = \frac{\mu_2}{\mu_3} - 1$$

The coefficients C_{1i} and C_{2i} are positive since all the terms in their expressions are positive hence equation (3.2) is a posynomial.

3.2.2 Constraints:

The following notation is used:

S_i = speed for the i-th operation, rpm,

$S_{ti\ max}$ = maximum speed limit for the i-th operation according to taylorized range+ and surface finish, rpm.

$S_{ti\ min}$ = minimum speed limit for the i-th operation according to taylorized range and surface finish, rpm,

$S_{m\ max}$ = maximum speed limit of machine tool, rpm,

$S_{m\ min}$ = minimum speed limit of machine tool, rpm,

f_i = feed for i-th operation, mm/rev.

$f_{ti\ max}$ = maximum feed limit for i-th operation according to taylorized range and surface finish, mm/rev.

+ Taylorized range is the range for which the Taylor's equation is valid.

$f_{m\ max}$ = maximum speed limit for machine tool, mm/rev.

$f_{m\ min}$ = minimum speed limit for machine tool, mm/rev.

The various constraints are represented as follows:

1) Maximum speed constraint

$$S_i \leqslant S_{i\ max} \qquad i = 1, \ldots, P \tag{3.3}$$

where $S_{i\ max} = \mathrm{Min}\ (S_{ti\ max}, S_{m\ max})$

2) Minimum speed constraint

$$S_i \geqslant S_{i\ min} \qquad i = 1, \ldots, P \tag{3.4}$$

where $S_{i\ min} = \mathrm{Max}\ (S_{ti\ min}, S_{m\ min})$

3) Maximum feed constraint

$$f_i \leqslant f_{i\ max} \qquad i = 1, \ldots, P \tag{3.5}$$

where $f_{i\ max} = \mathrm{Min}\ (f_{ti\ max}, f_{m\ max})$

4) Minimum feed constraint

$$f_i \geqslant f_{i\ min} \qquad i = 1, \ldots, P \tag{3.6}$$

where $f_{i\ min} = \mathrm{Max}\ (f_{ti\ min}, f_{m\ min})$

5) Power constraint

$$\sum_{i=1}^{P} H_i \leqslant H_{max} \tag{3.7}$$

where H_i = power requirement for i-th operation (HP)

H_{max} = maximum power available on machine tool (HP)

The force required for each operation [3] is given by,

$$F_i = K_2 \, d_i^{\mu_4} \, f_i^{\mu_5}$$

where K_2 is a constant and μ_4 and μ_5 are exponents of d_i and f_i respectively.

Therefore,

$$H_i = \frac{\pi \, D_i \, S_i \, F_i}{45000} \, (1/\eta) \tag{3.8}$$

where D_i = diameter of the i-th operation, mm

η = efficiency of transmission of power to the spindles

Eq. (3.8) is always positive with respect to the variables S_i and f_i, therefore constraint (3.7) is a posynomial.

Constraints (3.3) to (3.6) are also posynomials with coefficients of unity and exponents of unity, thus, all the constraints are posynomials.

3.3 PHASE 2: GROUPING OF OPERATIONS:

3.3.1 An Overview:

Since one of the objectives of cluster analysis is to obtain an insight into the underlying structure of the data or to find out 'natural groups' present in the data available, cluster analysis has been used for the operations grouping problem.

The obvious choice of characteristics to describe each operation are all data pertaining to speed and feed since these two are the control variables. Precedence (technological) constraints could be satisfied by including separate characteristics, for each operation, which would, in effect, keep two operations, which cannot be machined simultaneously, separate.

Further, the study has been made for 2 cases where the characteristics of the operations considered are:

1) maximum and minimum speed and feed limits, and precedence constraints only, and
2) optimal (Phase 1) speeds and feeds, and precedence constraints only.

If the first case (or both) provides 'good' solutions, we can group the operations prior to the optimization of cutting variables. However, if only the second case produces 'good' solutions then the optimization procedure of phase 1 has to precede grouping.

In the next section, the technique of cluster analysis is presented, while in section 3.3.3 the details of the procedure adopted for the grouping of operations are presented.

3.3.2 Cluster Analysis Technique:

3.3.2.1 Introduction:

In social, economic, industrial, psychological, biological and many other fields it is necessary to group similar items together, either to obtain an insight into the underlying structure of the data or to classify the items under study. The process of forming groups of similar items is called Cluster Analysis. Thus, given a data set, the objective of cluster analysis is to find out 'the natural groups' present in the data. The solution of a clustering problem should give the number of clusters present in the data and should list the members of each cluster. Obviously the members of one cluster should exhibit similarity with respect to a set of characteristics or features.

3.3.2.2 Mathematical Formulation:

Suppose that we have a set $O = [O_j \mid j = 1,2,\ldots, M]$ of M objects and another set $C = [C_i \mid i = 1,2,\ldots, N]$ of N characteristics (features) such that each characteristic (C_i) is a measurable feature of each object (O_j).

Let x_{ij} be the value of the i-th characteristic (C_i) of the j-th object (O_j), and let $X_j = [x_{ij}]$ denote the N x 1 vector of such measurements. Viewed geometrically, the M objects can be thought of as points in an N-dimensional space.

Given M objects and the data set $X = [X_j]$, a clustering algorithm will determine the natural clusters present in the data, such that objects belonging to the same cluster are 'highly similar'. Some 'criterion function' is used to evaluate the partitioning of objects.

3.3.2.3 Similarity Measure:

When do we say that the two objects i and j are similar? Now, since the M objects to be clustered can be represented as M points in an N-dimensional space, Euclidean distance is the most obvious (and also widely used) similarity measure. We keep two objects i and j in the same cluster if the distance between X_i and X_j is smaller than some specified threshold (D_1) and in different clusters if the distance is more than D_1. If the number of clusters is specified a priori then each object is assigned to the cluster it is closest to.

The Euclidean distance between any two points i and j in an N-dimensional space can be computed from the relationship:

$$d_{ij} = [\sum_{K=1}^{N} (x_{ik} - x_{jk})^2]^{\frac{1}{2}} \qquad (3.9)$$

where x_{ik} and x_{jk} are the values of the k-th characteristic of the i-th and j-th object respectively, and d_{ij} is the distance between them.

The difficulty in using Euclidean distance as a measure of similarity is that it is affected by change of scale. To overcome this drawback, we can standardise the variables (characteristics) by dividing the k-th characteristic by the maximum of all the k-th characteristics, i.e., normalization. The distance between two objects i and j then becomes,

$$d_{ij} = \left[\sum_{k=1}^{N} \left(\frac{x_{ik} - x_{jk}}{\max\limits_{\ell \in M} x_{\ell k}} \right)^2 \right]^{\frac{1}{2}} \tag{3.10}$$

3.3.2.4 Criterion Function:

In a clustering problem we want to partition a set $X = [X_1, X_2, \ldots, X_M]$ of M objects into exactly K clusters, where $K \leq M$, such that the objects belonging to the same cluster are similar (close to each other) and the objects belonging to different clusters are dissimilar. Now the number of ways of partitioning M objects into K distinct, nonempty clusters is given by Stirling's number of second kind [18],

$$W = \frac{1}{K!} \sum_{j=0}^{K} \binom{K}{j} (-1)^j (K-j)^M \tag{3.11}$$

(e.g., the total number of ways of partitioning 50 objects into 3 clusters is approximately equal to 2^{49}). Out of these W possible partitions we want to find a partition which optimizes a given objective function. The most commonly

used objective function in cluster analysis is the sum-of-squared-error criterion.

Let M_i be the number of objects in i-th cluster and Y_i be the mean vector of the objects belonging to the i-th cluster, so that

$$Y_i = \frac{1}{M_i} \sum_{X \in i} X \quad , \quad i = 1,2,\ldots, K \tag{3.12}$$

The mean vectors Y_1, Y_2, ..., Y_K are called 'centroids' and for a given cluster i, Y_i is the best representative of the objects belonging to i.

Now the sum-of-squared error in the i-th cluster is given by,

$$E_i = \sum_{X_j \in i} \left| X_j - Y_i \right|^2 \tag{3.13}$$

and the total sum-of-squared-error in all the K clusters is

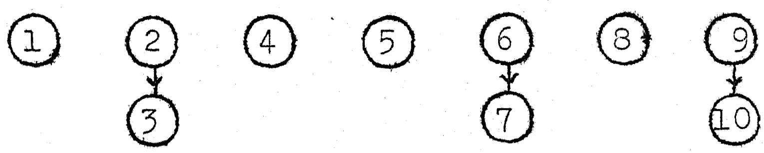
$$E = \sum_{i=1}^{K} E_i = \sum_{i=1}^{K} \sum_{X_j \in i} \left| X_j - Y_i \right|^2 \tag{3.14}$$

This criterion function has a simple interpretation. The sum-of-squared-error (E) measures the total squared error incurred in representing the M objects X_1, X_2, ..., X_M by K cluster centroids xY_1, Y_2, ..., Y_K [16].

3.3.3 Grouping of Operations Through Cluster Analysis:

The characteristics used to describe each operation are the maximum and minimum limits of feed and speed, and the optimal speed and feed selected in Phase 1. This choice is dictated by the fact that speed and feed are the control variables. The precedence or technological constraints which have to be represented as a characteristic are discussed next.

Consider the workpiece shown in Fig. 1, this job requires 10 turning operations (labeled from 1 to 10 in the figure). Of these 10 operations, some cannot be machined before others, viz., machining of operation 2 has to precede that of operation 3, operation 6 has to precede 7, and operation 9 has to precede 10. This can be represented diagrammatically as



where the numbers in the circles represent the operation numbers and the links represent precedence constraints, the arrowheads indicating the direction of precedence. Thus, for this job there is only one level of precedence, i.e., no operation is preceded by more than one operation. Further notice that the unlinked operations 1,4,5 and 8 can be grouped

with each other and any of the other operations, for instance, operation 1 can be grouped with operation 2 or operation 3. The similarity measure used is Euclidean distance so that all these above mentioned precedence features were represented by introducing an extra characteristic in the feature vector, called precedence value, such that if for operation 2 this value is 1 then the corresponding value for operation 3 will be 0, while that for operation 1 will be 0.5 because it should be equi-distant from 2 and 3.

The arrangement of the characteristics in the feature vector representing each operation is

$$(e_1, e_2, e_3, e_4, e_5, e_6, e_7)$$

where e_1 = maximum feed rate (mm/rev)

e_2 = minimum feed rate (mm/rev)

e_3 = maximum speed (rpm)

e_4 = minimum speed (rpm)

e_5 = optimum speed (rpm)

e_6 = precedence value (0 or 1 or 0.5)

e_7 = optimum feed (mm/rev)

The elements of the feature vectors of all operations are presented in Table 4. As indicated in Section 3.3.2.3 all the characteristics are normalized; but the precedence value is varied until the groups formed satisfy the precedence

constraints. This is done by multiplying the precedence values of all operations with a multiplication factor such that if the multiplication factor is 2, then, for example, the precedence values of operations 2, 1 and 3 will become 2.0, 1.0 and 0.0 respectively instead of the original values of 1.0, 0.5 and 0.0 respectively. The value of the multiplication factor which just satisfies all the precedence constraints is determined by trial and error.

Once the feature vectors are formed, groups (clusters) are formed using the clustering algorithm given by Mac Queen [1]. This algorithm is known as the K-means method [1] - nearest centroid sorting - where the number of groups K is specified a priori. In our study, K represents the number of spindles available for machining. The steps in the algorithm are as follows:

1) Take the first K feature vectors (operations) in the data set as groups of one member each.
2) Assign each of the remaining M-K, where M is the total number of feature vectors (operations), to the group with the nearest centroid. After each assignment, recompute the centroid of the gaining group.

3) After all the feature vectors (operations) have been assigned in step 2, take the existing group centroids as fixed seed points and make one more pass through the data set assigning each feature vector (operation) to the nearest seed point.

4) Repeat step 3 till convergence is achieved. (Three iterations were found to be sufficient for the test problem.)

Each grouping operation consists of three separate grouping operations as follows:

1) Grouping of the radial operations using the speed, feed and precedence elements.

2) Grouping of the axial operations using the speed, feed and precedence elements.

3) Grouping of the radial and axial groups found in steps (1) and (2) using only the speed and precedence elements. Moreover, the elements for this step are the centroids of the groups found in steps (1) and (2).

The reason for eliminating the feed elements in step (3) is that speed is the only common factor, other than precedence, between radial and axial operations as separate feeds are available for them on each spindle.

The FORTRAN IV code for the K-means algorithm given in [34] was modified to incorporate the Mac Queen algorithm

and the above procedure. The complete listing of the program is given in Appendix C.

In the next section the phase 3 which consists of optimization after grouping is presented.

3.4 PHASE 3 : OPTIMIZATION AFTER GROUPING:

The grouping of operations obtained in Phase 2 is used for the determination of optimal cutting parameters for individual groups of operations. The objective function and the constraints considered are listed in the following sections.

3.4.1 Objective Function:

The objective function is the minimization of cycle time (CT), which is dictated by the operation with the maximum processing time (PT). The PT of an operation is defined as the sum of the machining time for that operation and the average tool-change time per job for the tool concerned. The latter term having been included to keep a check on the number of stoppages of the machine tool caused by tool failures. Thus the objective function can be expressed as

$$\text{minimize} \quad CT = y$$

where

$$y = \max_{i} \left[\left(t_{mij} + t_{cij} \frac{t_{mij}}{T_{ij}} \right), \left(t_{mik} + t_{cik} \frac{t_{mik}}{T_{ik}} \right) \right] \qquad (3.15)$$

$$j = 1, 2, \ldots, q_{ai}$$
$$k = 1, 2, \ldots, q_{ri}$$
$$i = 1, 2, \ldots, N$$

All terms in Eqs. (3.15) have the same connotation as in eq. (3.1) except that

i indicates the spindle or group number,

j indicates the axial operation number, and

k indicates the radial operation number

Further,

N = total number of groups

q_{ai} = number of axial operations in the i-th group

q_{ri} = number of radial operationsin the i-th group

Since the terms in small parenthesis in Eqs.(3.15) are the same as for Eqs.(3.1) which have already been shown to have posynomial structure, the objective function in this phase also has a posynomial structure.

3.4.2 Constraints:

Let,

S_i = speed for the i-th group (or spindle)

$S_{ij\,max}$ = maximum speed for the j-th axial operation on i-th spindle

$S_{ij\,min}$ = minimum speed for the j-th axial operation on ith spindle

$S_{ik\ max}$ = maximum speed for the k-th radial operation on i-th spindle

$S_{ik\ min}$ = minimum speed for the k-th radial operation on i-th spindle

f_{ri} = feed for the radial slide on the i-th spindle

$f_{ik\ max}$ = maximum feed for the k-th radial operation on the i-th spindle

$f_{ik\ min}$ = minimum feed for the k-th radial operation on the i-th spindle

f_{ai} = feed for the axial slide on the i-th spindle

$f_{ij\ max}$ = maximum feed for the j-th axial operation on the i-th spindle

$f_{ij\ min}$ = minimum feed for the j-th axial operation on the i-th spindle

All speeds and feeds are in rpm and mm/revolution respectively.

The various constraints are represented as follows:

1) Maximum speed constraint

$$S_i \leqslant S_{i\ max} \tag{3.16}$$

where $S_{i\ max} = \text{Min}\ (S_{ij\ max}, S_{ik\ max}, S_{m\ max})$

$$i = 1, \ldots, N$$
$$j = 1, \ldots, q_{ai}$$
$$k = 1, \ldots, q_{ri}$$

2) Minimum speed constraint

$$S_i \geq S_{i\ min} \tag{3.17}$$

where $S_{i\ min} = \text{Max}\ (S_{ij\ min}, S_{ik\ min}, S_{m\ min})$

$i = 1, \ldots, N$

$j = 1, \ldots, q_{ai}$

$k = 1, \ldots, q_{ri}$

3) Maximum radial feed constraint

$$f_{ri} \leq f_{ri\ max} \tag{3.18}$$

where $f_{ri\ max} = \text{Min}\ (f_{ik\ max}, f_{m\ max})$ $\quad i = 1, \ldots, N$
$k = 1, \ldots, q_{ri}$

4) Minimum radial feed constraint

$$f_{ri} \geq f_{ri\ min} \tag{3.19}$$

where $f_{ri\ min} = \text{Max}\ (f_{ik\ min}, f_{m\ min})$ $\quad i = 1, \ldots, N$
$k = 1, \ldots, q_{ri}$

5) Maximum axial feed constraint

$$f_{ai} \leq f_{ai\ max} \tag{3.20}$$

where $f_{ai\ max} = \text{Min}\ (f_{ij\ max}, f_{m\ max})$ $\quad i = 1, \ldots, N$
$j = 1, \ldots, q_{ai}$

6) Minimum axial feed constraint

$$f_{ai} \geq f_{ai\ min} \tag{3.21}$$

where $f_{ai\ min} = \text{Max}\ (f_{ijmin}, f_{m\ min})$ $\quad i = 1, \ldots, N$
$j = 1, \ldots, q_{ai}$

7) Power constraint

$$\sum_{i=1}^{N} \left(\sum_{j=1}^{q_{ai}} H_{ij} + \sum_{k=1}^{q_{ri}} H_{ik} \right) \leqslant H_{max} \tag{3.22}$$

where H_{ij} = power requirement for the j-th axial operation on i-th spindle

H_{ik} = power requirement for the k-th radial operation on i-th spindle

The expression for power is the same as given by Eqs.(3.8).

8) The cycle time constraints

for axial operations

$$\left(t_{mij} + t_{cij} \frac{t_{kij}}{T_{ij}} \right) \leqslant y \tag{3.23}$$

$$i = 1, \ldots, N$$
$$j = 1, \ldots, q_{ai}$$

for radial operations

$$\left(t_{mik} + t_{cik} \frac{t_{mik}}{T_{ik}} \right) \leqslant y \tag{3.24}$$

$$i = 1, \ldots, N$$
$$k = 1, \ldots, q_{ri}$$

The constraints from Eqs. (3.16) to (3.24) have posynomial structure. The variables involved are either speed or feed, or both as in Eqs. (3.22) to (3.24).

3.5 SOLUTION METHODOLOGY:

The structure of the problems considered in phase 1 and phase 3 are posynomial, therefore Geometric Programming (GP) has been used.

The Blau algorithm [5] for GP was coupled with the Surrogate Geometric Programming (SGP) technique [58], because the Blau algorithm if used alone, consumes a large amount of time for computation owing to the high order of matrix inversion involved. For example, the optimization of cutting parameters for the problem shown in Fig. 1 involves inversion of matrices of order 62 and 39 for phases 1 and 3 respectively if the Blau algorithm is used as such. However, when coupled with SGP the order of the matrix to be inverted is reduced to 22 in phase 1 and to 11 in phase 3.

In the next two sections, i.e. Sections 3.6 and 3.7, the Blau algorithm and SGP technique will be presented .

3.6 THE BLAU ALGORITHM:

3.6.1 The General Format:

The most general primal form of a geometric program is as follows:

$$\text{Minimize} \quad y_o(\underline{X}) = \sum_{t=1}^{T_o} \sigma_{ot} \, C_{ot} \prod_{n=1}^{N} X_n^{a_{otn}} \tag{3.25}$$

$$\text{subject to} \quad y_m(\underline{X}) = \sum_{t=1}^{T_m} \sigma_{mt} \, C_{mt} \prod_{n=1}^{N} X_n^{a_{mtn}} \leqslant \sigma_m \, ; \tag{3.26}$$

$$m = 1, 2, \ldots, M \tag{3.27}$$

$X_n > 0; \quad n = 1, 2, \ldots, N$

where,

$\sigma_{mt} = \pm 1, \quad \sigma_{ot} = \pm 1$ and $\sigma_m = \pm 1$

$C_{mt} > 0, \quad C_{ot} > 0$

a_{mtn}, a_{otn} are unrestricted in sign

T_m = number of terms in the m-th constraint ($m = 1, 2, \ldots, M$)

T_o = number of terms in the objective function

In terms of engineering design formulations, the C_{mt} are economic coefficients, the X_n are design decision variables, the a_{mtn} are technological exponents of the decision variables, and the σ vector has as elements binary variables ($\pm$ 1), whose signs represent the sign of each term and inequality in the problem statement.

3.6.2 Basic Theory:

The interested reader is referred to [5] for the detailed theory but an explanatory extract is reproduced

here, 'The main idea is to use a Newton-Raphson procedure [Ref: Wilde and Beightler, 'Foundations of Optimization'] to drive to zero the components of the gradient of a Lagrangian function formed from the logarithms of the original objective function and the constraints. A non-linear transformation, which amounts to substituting a weighting variable for each term, makes the Lagrangian gradient linear in the weights as well as in the Lagrangian multipliers, although bilinear in the two sets of variables taken together. The polynomial form of all the functions makes available, in closed form, the derivatives needed for the Newton-Raphson iteration. To begin, one selects a trial primal solution, not necessarily satisfying the constraints exactly. This immediately gives the initial values of all the weight variables, which, when substituted into the Lagrangian gradient, gives a set of functions linear in the unknown Lagrangian multipliers. The initial values of multipliers are then chosen to minimize the sum of squares of these functions. Substitution of these values of weights and Lagrange multipliers into simple formulas gives the numbers needed for a Newton-Raphson adjustment, obtained by solving a non-singular set of linear equations. This gives new values of the Lagrange multipliers and of the original primal variables. When the latter are transformed into new weights, the Newton-Raphson procedure is repeated'.

The algorithm is reproduced in Appendix A

3.6.3 Users' Reports:

Two reports have been recorded in literature on the use of GOMTRY computer code. In the first, Hwner and Mehta [35] develop a queing model for a Computer I/O and C.P.U. overlap system and formulate two performance measures based on thruput and system utilization. These measures were optimized and validated by applying them to an existing system. The problem formulated had 2 variab one constraint and 10 terms.

In the second report, Nicoletti and Mariani [50] u the computer code to obtain numerical solutions to some general problems in management, modeled by discrete nonlin systems with nonlinear performance indices. Two examples discussed relate to 'structural control in a graded manpow system' and to 'advertising scheduling', the latter compris of 9 variables, 25 terms and 5 constraints.

Further, the code was tested satisfactorily on two of the test problems provided by Dembo [13], namely, Heat exchanger design problem (8 variables, 6 constraints and 19 terms) and the 3-stage membrane separation process (13 variables, 13 constraints and 53 terms).

3.7 SURROGATE GEOMETRIC PROGRAM:

3.7.1 Introduction:

A surrogate constraint is a linear combination of the constraint set. Thus, for a primal posynomial program given by eqs. (3.25), (3.26) and (3.27) with all σ's positive, the corresponding surrogated program is

$$\text{minimize} \qquad y_o\ (\underline{x}) \tag{3.28}$$

$$\text{subject to} \qquad \sum_{m=1}^{M} \gamma_m\ y_m\ (\underline{x}) \leq 1 \tag{3.29}$$

$$\text{and} \qquad x_n > 0\ ; \quad n = 1, \ldots, N \tag{3.30}$$

where γ_m are nonnegative scalars that are normalized by requiring

$$\sum_{m=1}^{M} \gamma_m = 1 \tag{3.31}$$

These γ_m will be referred to as <u>surrogate multipliers</u>. In order for the surrogated geometric program to be equivalent to the original program there must exist a set of variables $\underline{\gamma}^*$ such that the optimal solution $\underline{x}^*$ of Eqs. (3.28), (3.29) and (3.30) is optimal for the original primal problem. For proof of the existence and uniqueness of $\underline{\gamma}^*$ the reader is referred to the book by Phillips and Beightler [58].

The solution procedure involves searching possible values of $\underline{\gamma}$ for $\underline{\gamma}^*$. At each iteration of the procedure a value of $\underline{\gamma}$ will be chosen and the resultant S.G.P

problem solved. The solution that is obtained will then be checked for feasibility in the original problem and the chosen value for γ will be altered such that at each iteration it will better approximate the value of $\bar{\gamma}$.

3.7.2 Problem Reframed in S.G.P. Format:

The structure of the objective function remains the same while the constraints are reformulated as follows:

Phase 1 In the general GP format the constraints are expressed as

$$\frac{S_i}{S_{i\ max}} \leqslant 1 \ ; \quad \frac{S_{i\ min}}{S_i} \leqslant 1 \ ; \quad \frac{f_i}{f_{i\ max}} \leqslant 1 \ ;$$

$$\frac{f_{i\ min}}{f_i} \leqslant 1 \ ; \quad \text{and} \quad \frac{1}{H_{max}} \sum_{i=1}^{P} H_i \leqslant 1$$

$$i = 1, \ldots, P \tag{3.31}$$

Therefore the one constraint in S.G.P. format can be represented as

$$\sum_{i=1}^{P} \gamma_i \frac{S_i}{S_{i\ max}} + \sum_{i=1}^{P} \gamma_{P+i} \frac{S_{i\ min}}{S_i}$$

$$+ \sum_{i=1}^{P} \gamma_{2P+i} \frac{f_i}{f_{i\ max}} + \sum_{i=1}^{P} \gamma_{3P+i} \frac{f_{i\ min}}{f_i}$$

$$+ \frac{\gamma_{4P+1}}{H_{max}} \sum_{i=1}^{P} H_i \leqslant 1 \tag{3.32}$$

where $\sum_{i=1}^{P} (\gamma_i + \gamma_{P+i} + \gamma_{2P+i} + \gamma_{3P+i} + \gamma_{3P+i}) + \gamma_{4P+1} = 1$ (3.33)

Phase 3 Similarly the one constraint in S.G.P. format for Phase 3 can be represented as

$$\sum_{i=1}^{N} \gamma_i \frac{S_i}{S_{i\ max}} + \sum_{i=1}^{N} \gamma_{N+i} \frac{S_{i\ min}}{S_i} + \sum_{i=1}^{N} \gamma_{2N+i} \frac{f_{ai}}{f_{ai\ max}}$$

$$+ \sum_{i=1}^{N} \gamma_{3N+1} \frac{f_{ai\ min}}{f_{ai}} + \sum_{i=1}^{N} \gamma_{4N+i} \frac{f_{ri}}{f_{ri\ max}}$$

$$+ \frac{\gamma_{6N+1}}{H_{max}} \sum_{i=1}^{N} \left(\sum_{j=1}^{q_{ai}} H_{ij} + \sum_{k=1}^{q_{ri}} H_{ik}\right) + \sum_{i=1}^{N} \gamma_{5N+i} \frac{f_{ri\ min}}{f_{ri}}$$

$$+ \sum_{i=1}^{N} \left[\sum_{j=1}^{q_{ai}} \gamma_{ij} \frac{B_{ij}}{y} + \sum_{k=1}^{q_{ri}} \gamma_{ik} \frac{B_{ik}}{y} \right] \leqslant 1 \qquad (3.34)$$

where,

$$B_{ij} = \left(t_{mij} + t_{cij} \frac{t_{mij}}{T_{ij}}\right) \qquad j = 1, \ldots, q_{ai}$$

$$i = 1, \ldots, N$$

$$B_{ik} = \left(t_{mik} + t_{cik} \frac{t_{mik}}{T_{ik}}\right) \qquad k = 1, \ldots, q_{ri}$$

where

$$\sum_{i=1}^{N} (\gamma_i + \gamma_{N+i} + \gamma_{2N+i} + \gamma_{3N+i} + \gamma_{4N+i} + \gamma_{5N+i})$$

$$+ \sum_{j=1}^{q_{ai}} \gamma_{ij} + \sum_{k=1}^{q_{ri}} ik) + \gamma_{6N+1} = 1 \qquad (3.35)$$

3.7.3 Solution Procedure:

The steps that comprise the Surrogated Geometric Programming Algorithm (SGPA) are as follows:

1. Check if the problem can be formulated as a posynomial and all $\underline{x}_n^*$ are finite positive numbers. If not, SGPA cannot be applied.

2. Determine if at least one constraint is binding at optimality. If not the primal objective function can be treated as an unconstrained posynomial programming problem and solved.

3. Choose the initial values for the surrogate multipliers.

In SGPA a sequence of posynomial problems having one constraint (the surrogate constraint) will be solved. The coefficients of the terms in the constraint will be altered by changing the value of $\underline{\gamma}$ until an $\underline{x}^*$ is located which satisfies the original constraints. The surrogate multipliers used in the problem that yields this $\underline{x}^*$ are the required values of $\underline{\gamma}^*$.

Obviously it would be beneficial to choose an initial value for $\underline{\gamma}$ as close to $\underline{\gamma}^*$ as possible, thus minimizing the number of iterations required by SGPA. If no information is available with respect to which constraints are most likely to be binding, then it is reasonable to start with all

$\gamma_m = 1/M$, $m = 1, \ldots, M$. Using this starting value is simply an admission that there is presently no reason to assume any distinctions between the constraints.

If, however, some knowledge regarding the relative importance of the constraints is available in advance, better starting values for γ_m can be obtained. If a constraint, $y_m(\underline{x})$, is not binding, then $\gamma_m^* = 0$. Thus, if a constraint is not binding, it makes no contribution to defining the location of the optimum. If a constraint is thought to be particularly important then its surrogate multiplier should be given a large value, while the surrogate multiplier of a relatively insignificant constraint should be given a small initial value.

4. Solve the surrogated problem using a suitable optimisation procedure.

It should be noted that $\underline{x}_k^*$ determined at this stage are the optimal values of $\underline{x}$ only for the particular programming problem obtained when a given value is assigned to $\underline{\gamma}$; $\underline{x}_k^*$ is not necessarily the solution to the original posynomial program. In fact, these two quantities are equal only when $\underline{\gamma} = \underline{\gamma}^*$.

Each successive $\underline{\gamma}$ will hereafter be written as $\underline{\gamma}^1, \underline{\gamma}^2, \ldots, \underline{\gamma}^k$. The $\underline{x}^*$'s corresponding to these $\underline{\gamma}$'s will be denoted by $\underline{x}_1^*, \underline{x}_2^*, \ldots, \underline{x}_k^*$.

5. Calculate the values of the constraint.

Once the value of $\underline{x}_k^*$ has been determined for the (k+1)th choice of $\underline{\gamma}$, this value can be used to ascertain whether the constraints for the original problem are satisfied for this $\underline{x}_k^*$. If $\underline{x}_k^*$ satisfies the original constraint set, then $\underline{x}_k^* = \underline{x}^*$ and $\underline{\gamma}^k = \underline{\gamma}^*$. If $\underline{x}_k^*$ does not satisfy the constraint set, then new values for $\underline{\gamma}$ must be selected. When $\underline{x}_k^*$ does not satisfy the constraint set, it will be convenient to subdivide this set into three mutually exclusive and totally exhaustive subsets. Let

$$O \equiv [m | y_m (\underline{x}_k) > 1]$$

$$Q \equiv [q | y_q (\underline{x}_k) < 1]$$

$$R \equiv [r | y_r (\underline{x}_k) = 1]$$

If $\underline{x}_k^*$ is the optimal value of the primal variable vector for the original geometric program, then when the constraint set is evaluated,

$$O = \emptyset \tag{3.36}$$

when this occurs the SGPA terminates. If Eq. (3.36) is not satisfied the algorithm continues to step 6.

6. Alter the values of the surrogate multipliers.

If the value of $\underline{x}_k^*$ corresponding to a $\underline{\gamma}^k$ is such that $O \neq \emptyset$, then $\underline{\gamma}^k \neq \underline{\gamma}^*$ and a new value of $\underline{\gamma}$ must be chosen. It is desirable to choose this new value $\underline{\gamma}^{k+1}$ in such a

manner that $y_o(\underline{x}^*_{k+1}) \longrightarrow y_o(\underline{x}^*)$. First calculate the following value,

$$\alpha_m = A\,(\sum_{i\varepsilon Q} \gamma_i^k)\; y_m(\underline{x}_k) / \sum_{i=1}^{M} y_i(\underline{x}_k); \quad m\ \varepsilon\ O \quad (3.37)$$

Then calculate the new estimates of γ_m^{k+1} using the following equations:

$$\gamma_m^{k+1} = \gamma_m^k + \alpha_m ; \qquad m\ \varepsilon\ O$$

$$\gamma_m^{k+1} = (\gamma_{mL} + \gamma_{mH})/2 ; \qquad m\ \varepsilon\ R$$

$$\gamma_m^{k+1} = [\sum_{i\varepsilon Q} \gamma_i - \sum_{i\varepsilon P} \alpha_i + \sum_{i\varepsilon R} (\gamma_i^k - \gamma_i^{k+1})]/n(Q); \quad m\varepsilon Q$$

where,

A = a constant which may be used to speed up the convergence

γ_{mL} = last value of surrogate multiplier for which $y_m(\underline{x}) < 1$, $m\ \varepsilon\ R$.

γ_{mH} = last value of surrogate multiplier for which $y_m(\underline{x}) > 1$, $m\ \varepsilon\ R$

$n(Q)$ = number of terms in the set Q

Elements of set R are retained for 5 SGP iterations in order to avoid premature shift to set Q.

The multipliers are then normalized after equating to zero any multiplier with a negative value.

7. Repeat Steps 4 - 6.

Steps 4 - 6 are repeated until all the constraints are satisfied within some predetermined tolerance.

In the next chapter the results of the application of the solution methodology, given in this chapter, on a test case are presented and discussed.

CHAPTER IV

RESULTS AND DISCUSSION

The solution methodology suggested in Chapter III has been tested on a test problem available in literature [3]. The workpiece to be machined on a multi-spindle automatic has the characteristics shown in Fig. 1. The various input parameters pertaining to the multi-spindle automatic and the workpiece are tabulated in Table 1. The results of the test problem are presented in Section 4.2. However, before we discuss the results, a brief report regarding the computational experiences and difficulties encountered in the implementation of the methodology proposed in Chapter III will be presented.

4.1 IMPLEMENTATION:

4.1.1 Difficulties Encountered:

Initially the Blau algorithm was directly used as a solution procedure for Phase 1 of the methodology. It was observed that the program was getting terminated due to excessive overflow conditions. The cause of this problem was not immediately traceable. It was decided to use the S.G.P. approach which reduced the Newton-Raphson matrix of the Blau algorithm from an order of 62 to 22, so that the

time required for matrix inversions was greatly reduced. Moreover, the SGP approach rendered the problem more tractable and enabled detection of the earlier source of error which was as follows: the Blau algorithm requires a 'good' initial solution to start the procedure, but since the initial solution used was not 'good' (refer Table 2), the procedure for calculating the initial Lagrange multipliers, suggested by Blau, failed. This feature caused trouble in the S.G.P approach, too, at the initial stages. The source of error was detected after a study of two test problems given by Dembo [13], for which the magnitudes of the Lagrange multipliers were in the range of 0.2 to 2.0, while for the Phase 1 problem the magnitude was greater than 1000.0. Further, the conclusion that the procedure for calculating the initial Lagrange multipliers was at fault, was reinforced when convergence was achieved after approximating the Lagrange ~~m~~multiplier, for only the first iteration of S.G.P., by unity. This indicated that by using this arrangement/the first iteration of S.G.P. provided a 'good' initial solution for the second iteration. Though this arrangement provided convergence in Phase 3 also, it cannot be generalised to extend to other problems.

4.1.2 <u>S.G.P. - a Note [58]</u>:

It is interesting to note that S.G.P. algorithm provides useful information at every iteration. This

information is not obtained when the problem is solved using other techniques. Notice that at the k-th iteration, the problem,

$$\begin{aligned} &\text{minimize} \quad y_o(\underline{x}) \\ &\text{subject to} \quad y_m(\underline{x}) \leqslant y_m(\underline{x}_k) \ ; \quad m = 1, \ldots, M \\ &\qquad\qquad\qquad \underline{x} > 0 \end{aligned}$$

is solved. For finite k, it is the solution to a system that is the same as the original problem, given by eqs.(3.29), (3.30), except that the levels of resource availability have been changed. Hence it gives an optimal solution to a model that might be an acceptable alternative to the original model.

Another important by product of the SGP algorithm is that it provides information for sensitivity analysis. At any iteration the values of the surrogate multipliers for the problem that has been solved at that iteration are known. In particular, when the optimal value of the original program is found, the optimal values of the surrogate multipliers are known. These multipliers give the relative amount by which the optimal value of the objective function would change due to change in resource levels. Hence, the surrogate multipliers indicate which constraints are most binding and which constraints are of least importance.

Let us now turn our attention to Table 3. Presented in this table is the progress of the optimization algorithm (SGP) in terms of the constraints (which were tight or nearly tight at termination of the program), the corresponding surrogate multipliers and the objective function, for Phase 1. It has been presented to indicate the efficacy of the updating procedure for the surrogate multipliers, which has been developed for this thesis (a procedure suggested by Staats in [58] was found to proceed at a crawling pace). The first column of this table indicates the number of iterations of the Blau algorithm required for the corresponding S.G.P. iteration. The nomenclature for the symbols used is as follows:

y_0 is the magnitude of the objection function

$y_1, \ldots, y_{10}$ are the magnitudes of the upper feed limit constraints for operations 1 to 10 respectively

$y_{11}, \ldots, y_{20}$ are the magnitudes of the lower feed limit constraints for operations 1 to 10 resp.

$y_{21}, \ldots, y_{30}$ are the magnitudes of the upper speed limit constraints for operations 1 to 10 resp.

$y_{31}, \ldots, y_{40}$ are the magnitudes of the lower speed limit constraints for operations 1 to 10 resp.

y_{41} is the magnitude of the power constraint

and γ_i is the surrogate multiplier for constraint y_i, $i = 1, 2, \ldots, 41$.

Further a constraint value

= 1 indicates that the constraint is tight

< 1 indicates that the constraint is loose

> 1 indicates that the constraint is violated

Based on the above discussion on S.G.P., referring to Table 3 we find that at the 8th iteration of the S.G.P. algorithm, the power constraint is the most binding one (γ_{41} = 0.885 while the values for the other multipliers are below 0.02). Further the value of the surrogate multiplier indicates that the sensitivity of the objective function to the power constraint relative to the remaining constraints is 88.5 percent.

Now we shall discuss the results of the test problem.

4.2 RESULTS:

4.2.1 Phase 1:

The test problem has 10 operations. This results in 20 variables and 41 constraints. Corresponding to each operation there would be one variable for speed and another one for feed. For each variable there are two constraints corresponding to its upper and lower limits. Further, there would be a constraint resulting from the total power available on the machine tool.

Table 2 presents the initial solution selected for the cutting parameters. The initial feed values for each operation were chosen within the limits provided by the Taylorized range. However, the initial values for speeds were chosen near the lower speed limits, decided by the Taylorized range. This was done to avoid the violation of the power constraint. Table 2 also presents the optimal ~~xx~~ values of the cutting parameters obtained for each operation. It should be noted, with reference to Table 3, that for the optimal solution obtained two constraints are getting violated, those constraints are:

1. the feed constraint for operation 9. As y_9 is 1.005, the violation of the constraint is by 0.5 percent.
2. the power constraint. As y_{41} is 1.05, which implies that the violation of the constraint is by 5 percent.

These violations were not considered excessive. As a matter of fact, the criterion adopted was that all the constraint violation should be less than 5 percent.

The Phase 1 took a CPU time of 4 minutes on IBM 7044.

4.2.2 Phase 2:

The output of Phase 1 is used as the input for this phase. The feature vectors of the various operations are presented in Table 4. The results of the clustering are given in Tables 5(B) and 5(C).

Table 5(B) gives the results of grouping when only the minimum and maximum speed and feed limits and precedence values are considered in the feature vector; in the computer program this was accomplished by equating all the optimal speed and feed values of the feature vectors to zero before grouping. The multiplication factor for the precedence values, required to just satisfy the precedence constraints, is 1.22.

Table 5(C) gives the results of grouping when only the optimal feeds and speeds (of Phase 1) and precedence values are considered; this was accomplished in the computer program by equating the remaining elements of the feature vectors to zero before grouping. The required value of multiplication factor is 0.74.

The CPU time required on IBM 7044 in each case was 0.5 minutes.

4.2.3 Phase 3:

The grouping information obtained in Phase 2 considering optimal speeds and feeds and precedence values was used as input for this phase. The problem was formulated for a 3-spindle automatic which yielded 9 variables - 3 speeds, 3 axial feeds, and 3 radial feeds - and 29 constraints - 1 power constraint, 10 cycle time constraints, and 18 upper and lower limits for the 9 variables. The computer program was set to terminate when violation of all constraints

is below 5 percent. The results, optimal values of cutting parameters and the initial solution, are presented in Table 6. The initial values of the cutting parameters for a group were chosen arbitrarily from among the Phase 1 optimal values of that group. The cycle time obtained was 0.93 minutes. The constraints violated and the extent of violation are listed below:

1. Upper feed limit constraints for operations 3 and 9 were violated by approximately 0.3 percent, and that for operation 2 by 1 percent. But after rounding off the corresponding feed variables to 2 decimal places the constraints became tight.

2. The power constraint was violated by 4 percent.

The CPU time, on IBM 7044, required for this phase was 2.5 minutes.

4.2.4 Comparison of Results:

Two sets of results obtained by Bartalucci et al [3] pertaining to grouping of operations are presented in Table 5(A). These results pertain to 3 groups (or spindles). It is observed that the following operations occur together in both sets,

1. Operations 3, 7 and 10
2. Operations 2, 4 and 6
3. Operations 8 and 9
4. Operations 1 and 5

Taking these 4 groups of operations as the basis for comparison, we conclude that case C, i.e., the case in which only the optimal speeds and feeds (of phase 1) along with precedence relationships are considered, approximates the results of Bartalucci et al more closely.

The optimization results of phase 3 give a cycle time of 0.93 minutes and the total CPU time, on IBM 7044, required for all the three phases is 7 minutes. The corresponding results obtained by Bartalucci et al are presented in Table 7. From this table we observe that the cycle time of 0.93 minutes obtained in this study is very much less than the results obtained by Bartalucci et al. Moreover, the time required for the computation, using the 3-phase procedure, is much lower. Further, it should be noted that IBM 7090 (used by Bartalucci et al) is computationally faster than IBM 7044 by a factor of 1.4.

Table 8 represents the allocation of groups to the spindles of the 3-spindle automatic alongwith the type of tool and slide type for each operation.

4.3 SCOPE FOR FUTURE WORK:

The present work considered the determination of optimal cutting parameters for multi-spindle automatics without giving due emphasis on the evaluation and selection of alternate tool-change policies. However, it is a well-known fact that optimal cutting parameters and the choice of tool replacement policy are related, therefore, it would be worthwhile to develop models for the determination

of optimal tool replacement policy and optimal cutting parameters in an integrated fashion.

In the present work we have assumed the classical, deterministic Taylor's tool life equation. However, many investigators have reported in literature that tool life is a probabilistic phenomenon. Various types of distribution like Weibul, log-normal have been reported in literature [46, 51, 52, 53, 62, 63] for various types of tools. Further, Friedman and Tipnis [25] and Ravigiani et al [65] have presented a new perspective, viz., the Cutting-rate - Tool-life (R-T) characteristic curve which represents the loci of all optimal combinations of speed, feed and depth of cut, i.e., cutting rate, and the corresponding tool life values. The potential of this curve with respect to cutting parameter optimization merits investigation.

A new trend in manufacturing called Group Technology is assuming the proportions of a break-through and opens up a whole range of optimization problems for multi-spindle automatics. Group Technology could be introduced as a technique for identifying and bringing together related or similar components in a production process in order to take advantage of their similarities by making use of, for example, the inherent economies of flow production methods. Thus one needs

to develop models and methodologies for the adaptation of multi-spindle automatics to batch production environments considering Group Technology concepts.

Table 1 : Input Parameters

MACHINE TOOL: Total power = 40 HP, Efficiency (η) = 0.75, Min.Speed = 50.0 rpm, Max.Speed = 580.0 rpm.

WORKPIECE:

Opr.No.	1	2	3	4	5	6	7	8	9	10
Type	RADIAL	AXIAL	RADIAL	AXIAL	RADIAL	AXIAL	RADIAL	RADIAL	AXIAL	AXIAL
Dia.(mm)	225.0	250.0	250.0	280.0	255.0	150.0	150.0	115.0	80.0	90.0
Length (mm)	25.0	45.0	15.0	40.0	25.0	45.0	15.0	35.0	95.0	10.0
Depth of cut (mm)	2.0	3.0	5.0	3.0	3.0	3.0	15.0	2.0	3.0	2.0
Max.feed (mm/rev.)	0.5	0.5	0.1	0.7	0.5	0.7	0.2	0.5	0.5	0.35
Min.feed (mm/rev.)	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
Tool change time (min.)	1.0	1.0	2.0	1.0	1.0	1.0	2.0	1.0	1.0	1.0

The common factors for all the tools are as follows:

All tools are carbide

Constant in cutting force law = 238.0

Constant in Taylor's law = 80.0

Taylorized range (M/min.) = 180.0 - 30.0

Exponents in Taylor's law $\mu_1 = 0.16$, $\mu_2 = 0.42$, $\mu_3 = 0.14$

Exponents in cutting force law $\mu_4 = 0.74$, $\mu_5 = 0.98$

Table 2: Input/Output of Optimization:Phase 1.

Type	Opr. No.	Initial guess		Optimum values		Processing time (Optimal) (min.)
		Speed (rpm)	Feed (mm/rev.)	Speed (rpm)	Feed (mm/rev.)	
Radial	1.	60.0	0.25	106.0	0.4	0.59
Axial	2	60.0	0.2	123.0	0.37	0.99
Radial	3	60.0	0.1	215.0	0.09	0.78
Axial	4	60.0	0.35	70.6	0.58	0.98
Radial	5	60.0	0.25	85.6	0.4	0.73
Axial	6	70.0	0.35	111.5	0.54	0.76
Radial	7	60.0	0.1	99.0	0.19	0.79
Radial	8	120.0	0.25	198.6	0.35	0.5
Axial	9	120.0	0.25	235.0	0.5	0.8
Axial	10	120.0	0.2	237.0	0.18	0.23

All values of speeds are rounded off to one decimal place.
All values of feeds and processing times are rounded off to two decimal places.

Table 3: Progress of Optimization Algorithm (SGP): Phase 1

No. of iterations in GOMTRY	Iter. No. (SGP)	γ_2	γ_3	γ_9	γ_{23}	γ_{41}	y_o (min)	y_2	y_3	y_9	y_{23}	y_{41}
11	1	0.0244	0.0244	0.0244	0.0244	0.0244	1.032	1.904	2.315	1.673	2.228	7.299
5	2	0.047	0.052	0.044	0.051	0.111	3.422	1.04	1.21	0.932	1.219	2.228
4	3	0.142	0.163	0.034	0.163	0.315	7.204	0.616	0.691	0.695	0.690	1.141
6	4	0.056	0.064	0.017	0.064	0.293	4.083	0.866	1.096	0.854	1.04	1.879
4	5	0.029	0.0862	0.0037	0.085	0.637	6.47	0.75	0.85	0.823	0.84	1.182
3	6	0.016	0.046	0.0035	0.045	0.749	6.756	0.77	0.919	0.825	0.885	1.119
3	7	0.0088	0.026	0.0028	0.024	0.83	6.981	0.817	0.94	0.882	0.919	1.078
2	8	0.0048	0.016	0.0021	0.014	0.885	7.138	0.925	0.937	1.005	0.941	1.05

The nomenclature for this table is given in page 56.

Table 4: Input for clustering (Grouping) program: Phase 2.

Type	Opr. No.	Max. feed (mm/rev.)	Min. feed (mm/rev.)	Max. speed (rpm)	Min. speed (rpm)	Opt. speed (rpm)	Predecence value	Opt. feed (mm/rev.)
Radial	1	0.5	0.05	299,0	48.0	106.0	0.5	0.4
Axial	2	0.5	0.05	229.0	38.0	123.0	0.0	0.37
Radial	3	0.1	0.05	229.0	44.0	215.0	1.0	0.09
Axial	4	0.7	0.05	204.0	34.0	70.6	0.5	0.58
Radial	5	0.5	0.05	204.0	42.0	85.6	0.5	0.4
Axial	6	0.7	0.05	382.0	62.0	111.5	0.0	0.54
Radial	7	0.2	0.05	362.0	20.0	99.0	1.0	0.19
Radial	8	0.5	0.05	382.0	119.0	198.6	0.5	0.35
Axial	9	0.5	0.05	716.0	119.0	235.0	0.0	0.5
Axial	10	0.35	0.05	572.0	119.0	257.0	1.0	0.18

Table 5: Operation Groups.

A: Groups obtained by Bartalucci et al [3]

Group No.	1	2	3
3 Groups with 40 HP	3,7,10	2,4,6	1,5,8,9
3 Groups with 60 HP	3,7,10	2,4,6,1,5	8,9

B: Groups obtained using only maxima, minima of speeds and feeds and precedence values of feature vector. Multiplication factor necessary for precedence values = 1.22.

3 Groups with 40 HP	3,7	2,6,9, 1,5	8,4,10

C: Groups obtained using only optimal speeds and feeds and precedence values of feature vector. Multiplication factor necessary for precedence values = 0.74.

3 Groups with 40 HP	3,7,10	8,2,6,9	1,5,4

Values indicated in the table are operation numbers labelled as in Fig. 1.

Table 6: Input/Output of Optimization: Phase 3-for a 3-Spindle Automatic.

Group No.	Radial Opr.	Axial Opr.	Initial solution			Optimal values			Processing time for group (min.)
			Speed (rpm)	Radial feed (mm/rev.)	Axial feed (mm/rev.)	Speed (rpm)	Radial feed (mm/rev.)	Axial feed (mm/rev.)	
1	3,7	10	100	0.09	0.18	161	0.1	0.13	0.93
2	8	2,6,9	235	0.35	0.5	204	0.19	0.5	0.93
3	1,5	4	85	0.4	0.35	116	0.24	0.37	0.92

Values of speeds have been rounded off to the nearest integer.

Values of feeds and processing times have been rounded off to 2 decimal places.

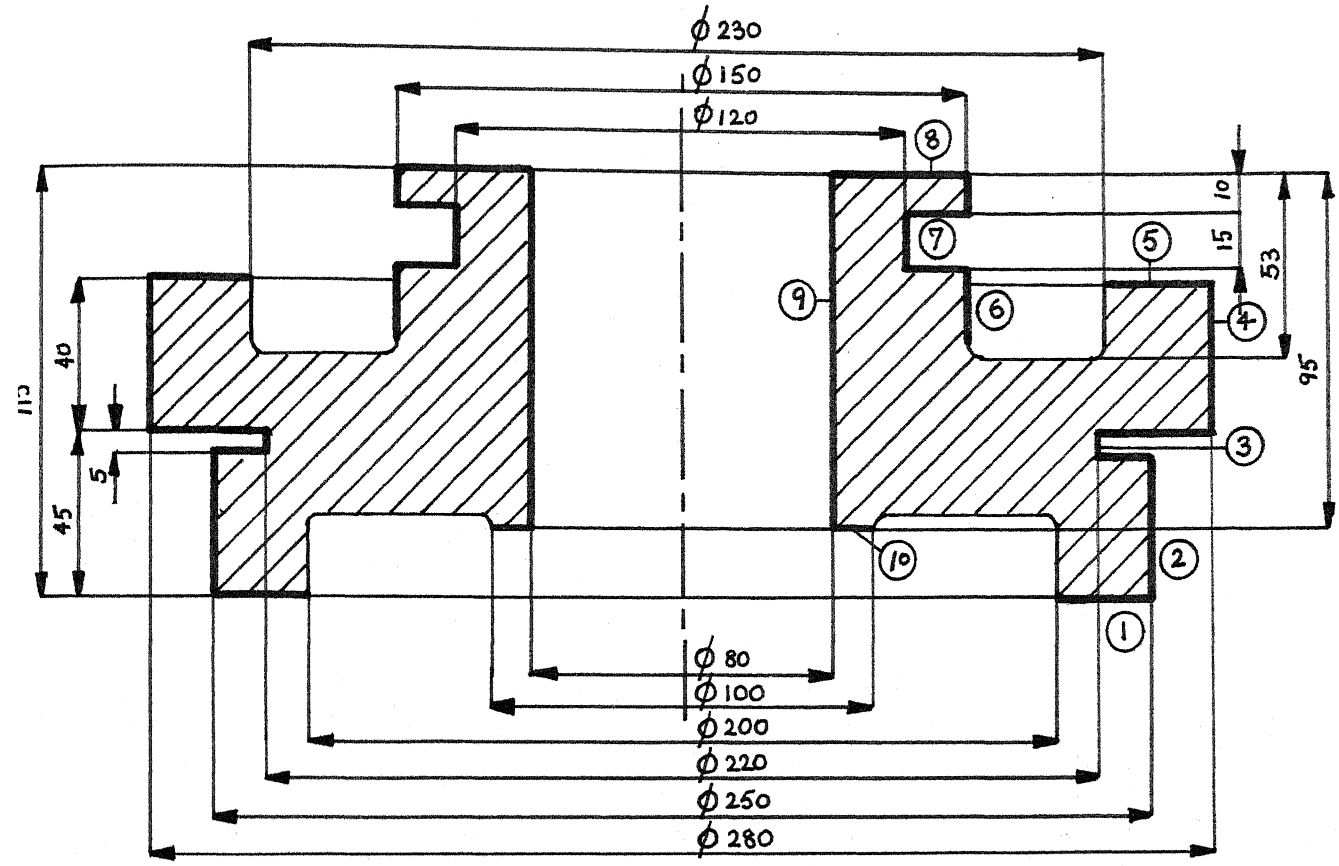
Cycle time corresponding to optimal results = 0.93 minute.

Table 7: Results obtained by Bartalucci et al

Number of spindles	Power (HP)	Cycle time (min.)	Computation time on IBM 7090 (min.)
5	60	1.51	28
3	60	1.54	8
3	40	2.42	8

Table 8: Allocations of Groups to Spindles.

Station No.	Group Allocated	Operations	Tool type	Slide type
1	2	2	Overhanging turning tool	Axial
		6	Turning tool	Axial
		9	Turning tool	Axial
		8	Facing tool	Radial
2	1	10	Recessing tool	Axial
		3	Flat form tool	Radial
		7	Flat form tool	Radial
3	3	4	Turning tool	Axial
		1	Facing tool	Radial
		5	Facing tool	Radial



All dimensions are in millimeters

Fig.1 : Test Problem : Job to be machined

REFERENCES

1. Anderberg, M.R., Cluster Analysis for Applications, Academic Press, 1973.

2. Armarego, E.J.A., and Russel, J.K., 'Maximum profit rate as criterion for the selection of machining conditions', Int. J. MTDR, Vol. 6, 1966, pp. 15.

3. Bartalucci, B., Bedini, B., and Lisini, G.G., 'Economics in Machining: Optimization on Automatic Lathes', S.M.E. Paper MR 70-235, 1971.

4. Bhattacharyya, A., Faria-Gonzalez, R., and Ham, I., 'Regression analysis for predicting surface finish and its application in the determination of optimum machining conditions', J.Engg. for Indus. Trans. ASME, Vol. 92, 1970, pp. 711.

5. Blau, G.E., and Wilde, D.J., 'Generalized Polynomial Programming', Canadian J. Chem. Engg., Vol. 47, 1969, pp. 317.

6. Brewer, R.C., 'On the economics of the basic turning operation', J.Engg. for Indus., Trans. ASME, Vol. 80, 1958, pp. 1479.

7. Brewer, R.C., 'Parameter selection problem in machining', Annals of CIRP, Vol. 14, 1966, pp.11.

8. Brewer, R.C. and Rueda, R., 'A simplified approach to the optimum selection of machining parameters', Engineer's Digest, Vol. 24, 1963, pp. 133.

9. Brown, R.H., 'On the selection of economical machining rates', Int. J. Prod. Res., Vol. 1, No. 2, 1962, pp.1.

10. Cook, N.H., Manufacturing Analysis, Addison-Wesley Publishing Co. Inc., 1966.

11. Crookall, J.R., and Venkataramani, A., 'Computer Optimization of Multipass turning', Int. J.Prod. Res., Vol. 9, No. 2, 1971, pp. 247.

12. De Vor, R.E., Anderson, D.L. and Zdeblick, W.J., 'Tool life variation and its influence on the development of tool life models', J.Engg. for Indus., Trans. ASME, Vol. 99, No. 3, 1977, pp.578.

13. Dembo, R.S., 'A set of Geometric Programming Test Problems and their solutions', Mathematical Programming, Vol. 10, 1976, pp. 192.

14. Dinkel, J.J., and Kochenberger, G.A., 'A Numerical procedure for sensitivity analysis in Geometric Programming', Engg. Optimization, Vol. 2, No. 3, 1976, pp. 41.

15. Doris, M.M., Savoye, G., and Megrelis, Reported by P. Weill, Proc. 3rd. Int. Conf. MTDR, 1962, pp. 279.

16. Duda, R.O., and Hart, P.E., Pattern classification and Scene Analysis, Wiley - Interscience, 1973.

17. Duffin, R.J., Peterson, E.L. and Zener, C., Geometric Programming, Wiley, New York, 1967.

18. Duran, B.S. and Odell, P .L., Cluster Analysis: A survey, Lecture Notes on Economics and Mathematical Systems, Vol. 100, Springer Verlag, 1974.

19. Edwards, G.A.B., Readings in Group Technology, The Machinery Publishing Co. Ltd., London, 1971.

20. Ermer, D.S., 'Optimization of the constrained Machining Economics Problem by Geometric Programming', J. Engg. for Indus., Trans. ASME, Vol. 93, No. 4, 1971, pp. 1067.

21. Esterzon, M.A., and Radzievskii, D.V., ' Group tool changes in transfer lines', Machines and Tooling, Vol. 39, No. 2, 1968, pp. 7.

22. Etin, A.O., et al , 'Tool selection for automatic machines', Russian Engg. J., Vol. 53, No. 12, 1973, pp. 43.

23. Field, M., Zlatin, N., Williams, R., and Kronenberg, M., 'Computerized determination and analysis of cost and production rates for machining operations, Part 1:, Turning, J.Engg. for Indus., Trans. ASME, Vol. 90, 1968, pp. 455.

24. Field, M., Zlatin, N., Williams, R., and Kronenberg, M., 'Computerized determination and analysis of cost and production rates for machining operations, Part 2: Milling, drilling, reaming and tapping', J.Engg.for Indus., Trans. ASME, Vol. 91, 1969, pp. 585.

25. Friedman, M.A., and Tipnis, V.A., 'Cutting-rate tool-life (R-T) characteristic functions for Material Removal Processes, Part I: Theory, Part II: Verification and Applications', J. Engg. for Indus., Trans. ASME, Vol. 98, 1976, pp. 481.

26. Fromm, F.R., and Northouse, R.A., 'CLASS: A non-parametric clustering algorithm', Pattern Recognition, Vol. 8, No. 3, 1976, pp. 45.

27. Gorina, T.I., et al , 'Simplex Method for Optimization of Cutting conditions', Russian Engg. J., Vol. 51, No. 10, 1971, pp. 59.

28. Gilbert, W.W., Machining -- Theory and Practice, American Society for Metals, 1950.

29. Gil'man, A.M., et al , 'Optimum conditions for metal cutting machine tools', Russian Engg. J., Vol. 48, No. 9, 1968. pp. 59.

30. Goranskii, G.K., Theory of Automation of Production Planning and of Tooling, Israel Program for Scientific Translations, Jerusalem, 1967.

31. Gracia, A., and Hogg, G., Computer code for Blau algorithm, University of Illinois at Urbana-Champaign, 1972.

32. Gurevich, D.M., 'Computerized Calculation of cutting conditions for N.C. lathes', Russian Engg. J., Vol. 49, No. 3, 1969, pp. 79.

33. Ham, I., and Faria-Gonzalez, R., ' Production Optimization Method (POM) using a digital computer', Advances in MTDR, Vol. A, Sept. 1970, pp. 493.

34. Hartigan, J.A., Clustering Algorithms, John Wiley and Sons, New York, 1975.

35. Hymer, W.A., and Mehta, R.P., 'Optimizing Computer Performance with Geometric Programming', European J. of Op.Res., Vol. 1, No. 2, 1977, pp. 103.

36. Iwata, K., Murotsu, Y., Iwatsubo, T., and Fujii, S., 'A probabilistic approach to the determination of optimum cutting conditions', J. Engg. for Indus., Trans. ASME, Vol. 94, No. 4, 1972, pp. 1099.

37. Iwata, A., Murotsu, Y., and Oba, F., 'Optimization of cutting conditions for multi-pass operations considering probabilistic nature of the machining process', J.Engg. for Indus. Trans. ASME, Vol.99, No. 1, 1977, pp. 210.

38. Jones, W.L., and Morgan, J.R., 'Use of decision diagrams to examine tooling problems', Int.J.Prod. Res., Vol. 6, No. 2, 1967/68, pp. 123.

39. Kapustin, N.M., et al , 'Optimizing operations for vertical multi-spindle automatics', Russian Engg. J., Vol. 53, No. 11, 1973, pp. 49.

40. Kapustin, N.M., et al , 'Dependability of Universal composite fixtures', Russian Engg. J., Vol. 52, No.3, 1972, pp. 36.

41. Kennedy, J.N., 'A review of some cluster analysis methods', AIIE Trans., Vol. 6, No. 3, 1974, pp.78.

42. Kolev, K.S., 'Graphical Method of calculating cutting speeds and feeds', Russian Engg. J., Vol. 46, No.10, 1966, pp. 77.

43. Kops, L., 'Economical Aspects of cutting speed selection when turning stepped parts', J.Engg.for Indus., Trans. ASME, Vol. 93, No. 4, 1971, pp. 1113.

44. Kothari, S.D., 'Optimization in Multi-spindle Automatics', Submitted in partial fulfilment of the requirement for the Degree of M.Tech. at IIT Kanpur, Sept. 1973.

45. Kuester, J.L., and Mize, J.K., Optimization Techniques with Fortran, McGraw-Hill, New York, 1973.

46. Malik, M.A.K., 'Average life of equipment subjected to compound failure', J.Engg. for Indus., Trans. ASME, Vol. 99, No. 2, 1977, pp. 431.

47. Malinovskii, A.Ya, 'Equipment for Group Technology', Russian Engg. J., Vol. 50, No. 1, 1970, pp.77.

48. Mancini, L.J., and Piziali, R.L., 'Optimal design of Helical Springs by Geometrical Programming', Engg. Optimization, Vol. 2, No. 1, 1976, pp. 73.

49. Melidov, I.E., 'Optimum flow-line machining of components in one-off production', Russian Engg. J., Vol. 52, No. 2, 1972, pp. 71.

50. Nicoletti, B., and Mariani, L., 'Generalized Polynomial Programming - Its approach to the solution of some management problems', European J. of Op. Res., Vol. 1, No. 4, 1977, pp. 239.

51. Pas'Ko, N.I., 'Mean life of multi-tool set-ups,' Russian Engg. J., Vol. 49, No. 2, 1969, pp. 63.

52. Pas'Ko, N.I., 'Optimization of Multi-tool set-ups with tool-life scatter', Russian Engg. J., Vol. 50, No. 9, 1970, pp. 82.

53. Pas'Ko, N.I., 'Establishing an Optimal tool replacement problem', Machines and Tooling, Vol. 47, No. 1, 1976, pp. 31.

54. Petropoulos, P.G., 'Optimal selection of machining rate variables by Geometric Programming', Int. J. Prod. Res., Vol. 11, No. 4, 1973, pp. 305.

55. Petrov, V.A., Flowline Group Production Planning, Business Publications, London, 1968.

56. Phillips, D.T., 'Geometric Programming with slack constraints and degrees of difficulty', AIIE Trans., Vol. 5, No. 1, 1973, pp. 7.

57. Phillips, D.T., and Beightler, C.S., 'Optimization in tool engineering using Geometric Programming', AIIE Trans., Vol. 8, No. 2, 1970, pp. 142.

58. Phillips, D.T., and Beightler, C.S., Applied Geometric Programming, John Wiley and Sons, New York, 1976.

59. Radford, J.D., and Richardson, D.B., 'Optimization of parameters in turning', Production Engineer, Vol. 49, 1970, pp. 197.

60. Ragsdell, K.M., and Phillips, D.T., 'Optimal Design of a Class of Welded Structures using Geometric Programming', J. Engg. for Indus., Trans. ASME, Vol. 98, No. 3, 1976, pp. 1021.

61. Rajagopalan, R., Design Retrieval and Group Layout Through Fuzzy Sets and Graphs, Ph.D. Thesis, Industrial and Management Engineering Programme, IIT Kanpur, India, Dec. 1975.

62. Ramalingam, S., 'Tool-life distributions - Part 2: Multiple-injury tool life models', J. Engg.for Indus., Trans. ASME, Vol. 99, No. 3, 1977, pp. 523.

63. Ramalingam, S., and Watson, J.C., 'Tool life distributions - Part 1: Single-injury tool-life model', J.Engg. for Indus., Trans. ASME, Vol. 99, No. 3, 1977, pp. 519.

64. Rao, P.V., Kumar, S.N., and Janakiraman, P.A., 'Geometric Programming: A new hybrid algorithm based on Liapunov concepts', Engg. Optimization, Vol. 3, No. 2, 1978, pp. 93.

65. Ravignani, G.L., Tipinis, V .A., and Friedman, M.Y., 'Cutting-rate tool-life function (R-T-F): General Theory and applications', Annals of CIRP, Vol. 26, 1977, Pp. 295.

66. Reed, W.J., 'Automated Process Planning takes a giant step', Machine and Tool Blue Book, Vol. 72, No. 5, 1977, pp. 116.

67. Reed, W.J., 'CAPP catapults part family manufacturing to new levels', Machine and Tool Blue Book, Vol. 71, No. 3, 1976, pp. 76.

68. Raznikov, N.I., 'Calculating optimum speeds and feeds by computer', Russian Engg.J., Vol. 46, No. 12, 1966, pp. 61.

69. Romanov, K.F., and Sotnikova, K.F., 'Effect of workshop conditions on cutting speeds', Russian Engg.J., Vol. 46, No. 4, 1966, pp. 62.

70. Shats, A.S., and Nebylitskii, F.I., 'Adjustable Universal composite fixtures', Russian Engg. J., Vol. No. 2, 1970, pp. 74.

71. Sheparov, V.I., 'Group Tool changes in transfer lines', Machines and Tooling, Vol. 43, No. 1, 1972, pp. 44.

72. Sheparov, V.I., 'Graphical method of planning optimal tool changes', Machines and Tooling, Vol. 43, No.7, 1972, pp. 52.

73. Tilslay, R., Lewis, F.A., and Galloway, D.F., 'Flexible Cell Production Systems - A Realistic Approach', CIRP Annals, Vol. 26, No. 1, 1977, pp. 269.

74. Vul'fson, A., and Derybin, A.L., 'Speeds and feeds algorithm for programme-controlled lathes', Russian Engg. J., Vol. 46, No. 4, 1966, pp. 68.

75. Walvekar, A.G., and Lambert, B.K., 'An application of Geometric Programming to machining variables selection', Int. J.Prod. Res., Vol. 8, No. 3,1970, pp. 241.

76. Wu, S.M., and Ermer, D.S., 'Maximum profit as criterion in the determination of the optimum cutting conditions', J.Engg. for Indus., Trans. ASME, Vol. 88, 1966, pp. 435.

APPENDIX A

BLAU (GOMTRY) ALGORITHM

This algorithm finds the minimum of a multivariable, nonlinear function of geometric form represented by eqns. (3.25), (3.26) and (3.27). The particular algorithm included in this section was developed by Blau [5] and programmed by Garcia and Hogg [31]. The notations used are the same as for eqns. (3.25) to (3.27). The method proceeds as follows:

1) Enter problem with input parameters.

2) Determine initial weights:

$$Z = \sum_{t=1}^{T_o} \sigma_{ot} C_{ot} \prod_{n=1}^{N} x_n^{a_{otn}}$$

Dual function, $V = |Z|$

$$\beta_{mt} = C_{mt} \prod_{n=1}^{N} x_n^{a_{mtn}} \qquad n = 0,1,\ldots, M$$

$$\beta_{ot} = \beta_{ot}/V$$

3) Calculate the vector of orthogonal conditions for the objective, and the matrix of these conditions for the constraints:

$$K = [\sum_{t=1}^{T_m} \sigma_{mt}\, a_{mtn}\, \beta_{mt}] \qquad \begin{array}{l} m = 1,\ldots, M \\ n = 1,\ldots, N \end{array}$$

$$H_n = \sum_{t=1}^{T_o} \sigma_{ot}\, a_{otn}\, \beta_{ot} \qquad n = 1,\ldots, N$$

4) Evaluate initial Lagrange multipliers

$$\gamma = (K^T K)^{-1} K^T H \quad (K^T \text{ denotes } K \text{ transpose})$$

5) If this is the first iteration, go to Step 6); otherwise, determine new weights as in Step 2), new orthogonal conditions as in Step 3), and new multipliers, as follows:

$$\gamma_{new} = \gamma_{old} + \gamma$$

Now proceed to Step 6).

6) Calculate the matrix T

$$T = \left[\begin{array}{l} \sum_{m=1}^{M} \left[\sum_{t=1}^{T_m} \sigma_{mt}\, a_{mti}\, a_{mtj}\, \beta_{mt} \right] \gamma_m \\ \quad - \sum_{t=1}^{T_o} \sigma_{ot}\, a_{mti}\, a_{mtj}\, \beta_{ot} \end{array}\right] \qquad \begin{array}{l} i = 1,\ldots, N \\ j = 1,\ldots, N \end{array}$$

7) Evaluate error

$$e_i = \left[\begin{array}{ll} \sum_{t=1}^{T_o} \sigma_{ot}\, a_{oti}\, \beta_{ot} - \sum_{m=1}^{M} \left[\sum_{t=1}^{T_m} \sigma_{mt}\, a_{mti}\, \beta_{mt} \right] \gamma_m & i = 1,\ldots, N \\ \sigma_o - \sum_{t=1}^{T_o} \sigma_{ot}\, \beta_{ot} & i = N+1 \\ 1 - \sum_{t=1}^{T_m} \sigma_{mt}\, \beta_{mt} & i = N+1+m;\ m=1,\ldots,M \end{array}\right.$$

$$
\begin{array}{cc}
 & \begin{array}{ccc} i:\ 1,\ 2,\ \ldots,\ N & N+1 & N+2,\ N+3,\ \ldots,\ N+M+1 \\ n:\ 1,\ 2,\ \ldots,\ N & 0 & m:\ 1,\ 2,\ \ldots,\ M \end{array} \\
\begin{array}{c} J:\ 1,\ 2,\ \ldots,\ N \quad (n:\ 1,\ 2,\ \ldots,\ N) \\ N+1 \quad (0) \\ N+2,\ N+3,\ \ldots,\ N+M+1 \quad (m:\ 1,\ 2,\ \ldots,\ M) \end{array} & \left[\begin{array}{c|c|c} T & \bar{\bar{H}} & K \\ \hline \bar{\bar{H}}^T & +1 & \bar{0} \\ \hline K^T & \bar{0} & 0 \end{array}\right]
\end{array}
$$

9) Invert the matrix R.

10) Find vector of adjustments:

$$R^{-1}\bar{c} = \begin{bmatrix} \Delta \ln \bar{x} \\ \Delta \ln V \\ \Delta \ln \gamma \end{bmatrix}$$

11) Calculate new values of independent variables:

$$\bar{x} = \bar{x} \exp (\Delta \ln \bar{x})$$
$$V = V \exp (\Delta \ln V)$$

12) Has solution converged to acceptable limit?

<u>Yes</u>; print results and stop.

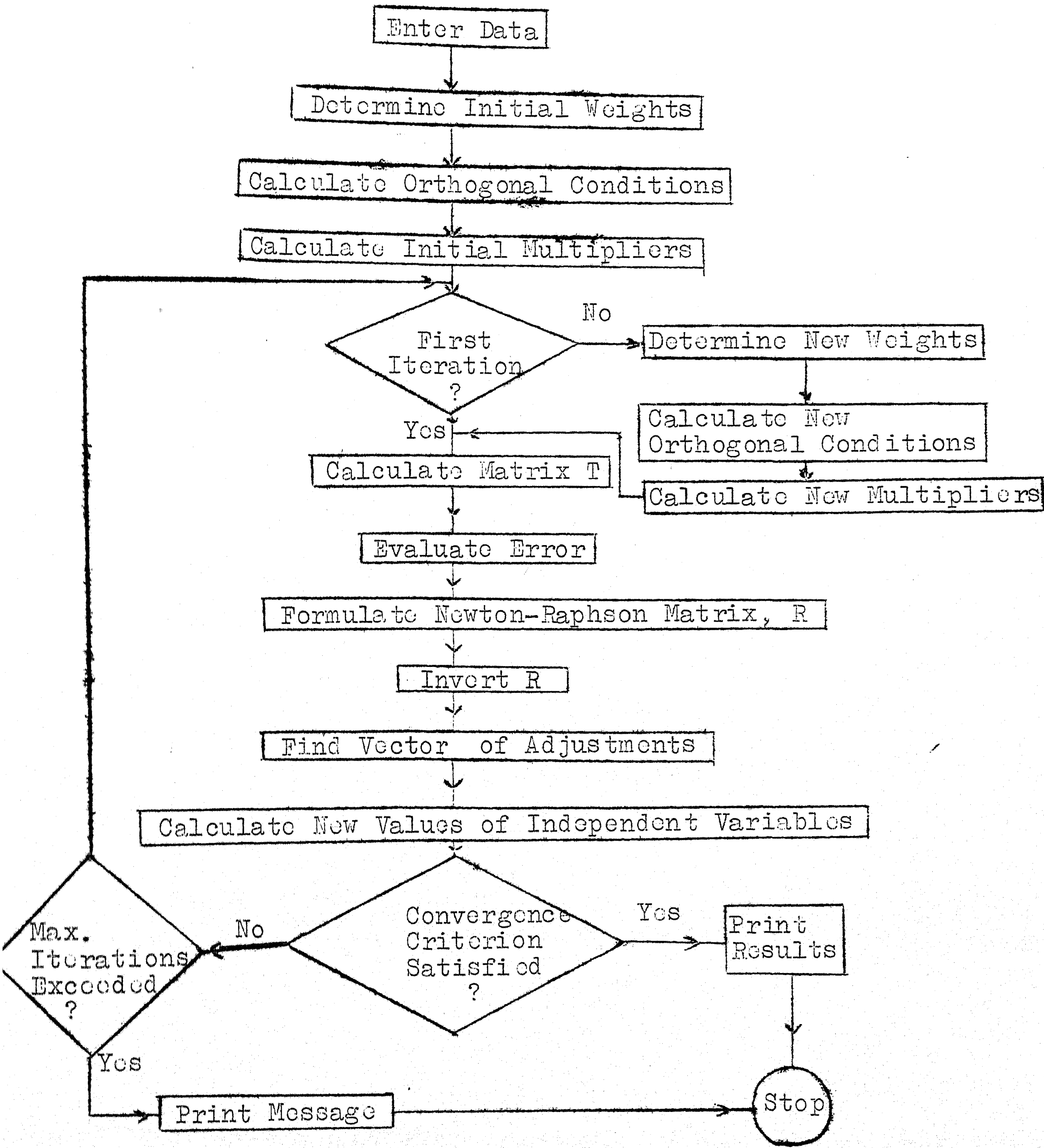
<u>No</u>; go to Step 13.

13) Has maximum allowable iterations been reached?

<u>Yes</u>; stop, print message.

<u>No</u>; return to Step 5 .

A flow diagram illustrating the above procedure is shown in next page.



Geometric Programming (GOMTRY ALGORITHM)
Logic Diagram.

APPENDIX B

SURROGATE GEOMETRIC PROGRAM

Usage:

The program consists of a main program and three subroutines, GP3, GP10, and GP22. The main program reads the data, prints the results, and performs a major part of the algorithm.

Subroutines Required:

Subroutine GP3 calculates the absolute value of all terms in the problem. Subroutine GP10 inverts the Newton-Raphson matrix, or one of its submatrices. Subroutine GP22 calculates orthogonal conditions for the objective function and the constraints. All linkage between main program and the three subroutines is through COMMON statements.

Description of Parameters:

N	Total number of variables
M	Number of constraints
NC	Number of constraints in the original Geometric Program
TS	Assumed sign of the optimal value of the objective function
MRX	Dimension of matrix R; (M+N+1)
CONVG	Convergence criterion

KT	Vector giving number of terms per polynomial
C	Vector of coefficients
A	Array of exponents
X	Vector of initial estimates of optimal solution
W	Vector of weights (dual variables)
E	Error vector
AMD	Vector of Lagrange multipliers
PIVOT	Vector used in matrix inversion
LY	Card reader unit number
LZ	Printer unit number
SUM	Objective function
V	Dual function
KTT	Total number of terms in the problem
M1	Total number of polynomials in the original Geometric Program.
SUR	Vector of surrogate multipliers

DIMENSION Requirements:

The DIMENSION statements should be modified according to the requirements of each particular problem. The parameters in parentheses included in the following statements conform to the Parameter definitions above:

```
DIMENSION      C(KTT), X(N), KT(M1), W(KTT), A(KTT,N),
               R(MRX, MRX), PIVOT (MRX), AMD (M), E(KTT),
               IPIVOT (MRX), INDEX (MRX, 2), CC (KTT),
               SUR (NC)
```

DIMENSION K(NC), WEX (NC), KTN (NC), NA(NC), MA(NC), LA(NC), SO(NC), STO(NC)

Input Formats:

Card type	FORMAT	CONTENTS
1	(2I4, F5.1)	N, M, TS
2	(2I4)	NC, MRX
3	(I4)	KT(L), L = 1, M1

(One type 3 card for each polynomial, including objective function)

4	(5E 15.8)	C(I) (Coefficient Ith term)

(If I is greater than 5, additional Type 4 cards are required)

5	(5E 15.8)	A(I,J) (Exponent for Jth variable in Ith term)

(If J is greater than 5, additional Type 5 cards required for each term.)

6	(F 11.5)	X(J), J = 1, N

(One type 6 card required for each variable)

Output:

The problem statistics which include number of variables, number of constraints, number of terms, and sign of the objective function are printed. Thereafter, on each S.G.P. iteration, the current solution, absolute value of all the terms and values of the surrogate

multipliers are printed. This is continued until the convergence criterion has been satisfied, after which the message 'CONVERGENCE CRITERION SATISFIED' is printed.

Summary of User Requirements:

1. Formulate problem according to requirements explained in Section 3.7.
2. Specify values for N, M, TS, NC, MRX, LZ, LY, CONVG.
3. Specify initial solution, X(J).
4. Adjust DIMENSION statements in main program and subroutines.
5. Modify FORMAT statements 1000, 1070, 1100 and their associated READ and WRITE statements for the particular problem being solved.

APPENDIX C

PROGRAM FOR GROUPING OF OPERATIONS

Usage:

The program consists of a main program and five subroutines BUILD, KMEANS, SINGLE, MIN and MAX. The main program reads the data and prints the results.

Subroutines Required:

Subroutine MIN finds the minimum of a set of values. Subroutine MAX finds the maximum of a set of values. Subroutine SINGLE updates the statistics for each group. Subroutine KMEANS determines the closest group for each operation. Subroutine BUILD initializes the statistics of each group and calls KMEANS. All linkage between the main program and the subroutines is through ar uments of the CALL statements.

Description of Parameters:

K	Number of spindles
M	Number of operations
N	Number of characteristics for each operation
ITER	Number of iterationsfor convergence of the Mac Queen procedure
ENC	Increment or decrement for the divisor which adjusts the procedence values ($-1 < ENC < 1$)

W	Vector of maximum values of all characteristics
WR	Vector of maximum values of all characteristics for radial operations
WA	Vector of maximum values of all characteristics for axial operations.
A	Array of characteristics for all operations
TIM	Vector of cutting times for all operations
OLD	Vector of cutting lengths of all operations
MAS	Vector of consisting of binary values: 0 indicates a radial operation and 1 indicates an axial operation
NAS	Vector of operation numbers (corresponding to Fig.1)
DCLUS	Vector of distances from group centroid

<u>DIMENSION Requirements</u>:

The DIMENSION statements should be modified according to the requirements of each particular problem. The parameters in parentheses included in the following statements conform to the parameter definitions above:

```
DIMENSION      WR(N), WA(N), TIM(M), OLT(M), FD(K,M),
               MCLU (K,M), A(M,N), W(N), SUM (8,N,K),
               X(N), NCLUS(M),  DCLUS (M)
DIMENSION      MA(M), MAS(M), NA(M), NAS(M), B(M,N), DX(M),
               JCLUS (M), X1(M), X2(M), X3(M), X4(M),NST(M)
```

Input Formats:

Card Type	Format	Contents
1	(7 F12.4,13X I 1, I3)	(A(I,J), J=1,N), TIM(I), OLT(I),MAS(I), MAS(I) (Values corresponding to the Jth characteristic of the Ith operation) (One card is required for the Ith operation, I = 1, M) W(J), J = 1, N WR(J), J = 1, N WA(J), J = 1, N ENC

Output:

The information in the print-out includes the group numbers, members of the group and maximum and minimum speed and feed limits for each group. The multiplication factor for the precedence values is also printed. All this information is repeated for 5 different values of multiplication factor.

Summary of User Requirements:

1. Formulate problem according to requirements explained in Section 3.3.3.
2. Specify values for K, M, N, ITER in the DATA statement.

3. Multiplication factor is given by [1/(V(6) - ENC)] so specify value of ENC so as to obtain required range for precedence values.

4. Adjust DIMENSION statements in main program and subroutines.

5. Modify FORMAT statement 2 and the associated READ statements for the particular problem being solved.

```
C-----------------------------------------------------------------------
C*****SOLUTION TO SURROGATE GEOMETRIC PROGRAMMING PROBLEM
C-----------------------------------------------------------------------
C
      DIMENSION C(70),X(20),KT(2),W(70),A(70,20),E(22,22),PIVOT(22),
     1AMD(1),E(70),IPVOT(22),INDEX(22,2),CC(70),SUR(41)
      DIMENSION K(41)
      DIMENSION WEX(41), KTN(41)
      DIMENSION MA(40), MA(40), IA(40), SO(41), STO(41)
      COMMON W,PIVOT,E,P,SUM,V,SIG,AMD,C,X,KT,A,MM,N,M,M1,KTT,KTO,MR,
     1E1,IT,EE1,NR
      KOU = 0
      MOT = 0
      N2 = 0
      LZ = 6
      LU = 0
      LY = 5
      WRITE (LZ,1120)
C***** INPUT TO PROBLEM
      WRITE (LZ,1110)
      READ (LY,1000) N,M,TS
      READ (LY,1000) NC, MEX
      S = TS
      SIG = TS
      M1 = M + 1
      READ (LY,1070) (C(I), I=1,KTT)
      DO 7 I=1,41
      K(I) = 0
    7 KTN(I) = 0
C***** NUMBER OF TERMS IN EACH POLYNOMIAL
      KTT = 70
      READ(LY,1080) (KT(I), I=1,M1)
C***** INITIALIZE SURROGATE CONSTANTS
      AT = 1./FLOAT(NC)
      DO 10 I=1,NC
   10 SUR(I) = AT
      KSIG = SIG
      IF (M-1) 15,15,20
   15 WRITE (LZ,1020) N,M,KTT,KSIG
      GO TO 30
   20 WRITE (LZ,1010) N,M,KTT,KSIG
   30 WRITE (LZ,1030)
      WRITE (LZ,1040)
C***** COEFFICIENTS AND EXPONENTS
      READ(LY,1070) (C(I), I=1,KTT)
      I2 = 0
      DO 32 L=1,M1
      I1 = I2 + 1
      I2 = I1 + KT(L) - 1
   32 READ (LY,1070) ((A(I,J), I=I1,I2), J=1,N)
C***** STORE COEFFICIENTS
      DO 34 I=1,KTT
   34 CC(I) = C(I)
C***** INITIAL SOLUTION
      DO 50 J=1,N
```

```
   50  READ (IY,1130) X(J)
       MR = M + N + 1
       M1 = M + 1
       MP = M + 2
       MR1 = MR + 1
       KTO = KT(1)
C***** REGULAR GOMTRY ALGORITHM STARTS HERE
C***** SURROGATE LOOP STARTS HERE
C***** UPDATE COEFFICIENTS
   52  DO 55 I=1,40
       II = 20 + I
   55  C(II) = CC(II) * SUR(1)
       DO 56 I=61,KTT
   56  C(I) = CC(I) * SUR(41)
       IT = 1
       DO 60 JR=1,MRX
       DO 60 IR=1,MRX
   60  P(IR,JR) = 0.
C***** EVALUATION OF INITIAL WEIGHTS
C***** THESE WEIGHTS WILL CHANGE AT EACH ITERATION
       CALL GP3
C***** EVALUATION OF THE OBJECTIVE FUNCTION
       SUM = 0.
       DO 70 I=1,KTC
   70  SUM = SUM + W(I) *(C(I)/ ABS(C(I)))
C***** INITIAL VALUE OF V
       V =  ABS(SUM)
       DO 80 I=1,KTC
   80  W(I) = W(I)/V
C-----------------------------------------------------------------------
 1000  FORMAT (I4,I4,F5.1)
 1010  FORMAT (1H0,9X,I3,10H VARIABLES,7X,I3,14H RESTRICTIONS ,7X,I3, 9H
      1TERMS     ,7X,14HOBJECTIVE SIGN ,I3)
 1020  FORMAT (1H0,9X,I3,10H VARIABLES,7X,I3,14H RESTRICTIONS ,7X,I3, 9H
      1TERMS     ,7X,14HOBJECTIVE SIGN ,I3)
 1030  FORMAT (1H0,37X,26HCOEFFICIENTS AND EXPONENTS  )
 1040  FORMAT (1H0,1X,4HTERM,2X,11HCOEFFICIENT, 4X, 4HX(1) ,5X,4HX(2),5X,
      14HX(3),5X,4HX(4),5X,4HX(5), 5X,4HX(6),5X ,  4HX(7),5X,4HX(8) )
 1050  FORMAT ( 5(I3,E17.8))
 1051  FORMAT (8F3.0)
 1070  FORMAT (5E15.8)
 1071  FORMAT (E15.8)
 1080  FORMAT (I4)
 1090  FORMAT (*   WEIGHTS*)
 1091  FORMAT (*   LAGRANGE MULTIPLIERS*)
 1092  FORMAT (*   ERROR VECTOR*)
 1093  FORMAT (*   PIVOTS*)
 1100  FORMAT (F11.5)
 1110  FORMAT (20X,45H-----   GEOMETRIC PROGRAMMING PROBLEM   -----  )
 1120  FORMAT (10X)
C-----------------------------------------------------------------------
       CALL GP22
C***** PRODUCT OF THE TRANSPOSE MATRIX OF ORTHOGONAL CONDITIONS
C***** FOR THE CONSTRAINTS BY THE SAME MATRIX WITHOUT TRANSPOSING
       DO 90 JR=NR,MR
```

```
      DO 90 KR=1,N
   90 R(IR,JR) = R(IR,JR) + R(KR,IR) * R(KR,JR)
      MM = NR
C***** INVERSION OF THE MATRIX FOUND ABOVE
      CALL GP12
C***** PRODUCT OF THE TRANSPOSE MATRIX OF ORTHOGONAL CONDITIONS
C***** FOR THE CONSTRAINTS BY THE VECTOR OF THE SAME CONDITIONS
C***** FOR THE OBJECTIVE FUNCTION
      DO 100 JR=NR,MR
      DO 100 IR=1,N
  100 R(N1,JR) = R(N1,JR) + R(IR,JR) * R(IR,N1)
C***** INITIAL MULTIPLIERS
      DO 110 IR=NR,MR
      L = IR - N1
      AMD(L) = 0.
      DO 110 JR=NR,MR
      AMD(L) = AMD(L) + R(N1,JR) * R(IR,JR)
C***** FOR THE FIRST SURROGATE LOOP LAGRANGE MULTIPLIER IS
C***** MADE A CONSTANT
      IF (NOT .EQ. 0)  AMD(1) = -1.
  110 CONTINUE
C***** OPTIMIZATION SUB-LOOP STARTS HERE
      DO 310 KKK=1,70
      IF (IT - 1)  155,115,155
C***** FIRST ITERATION IS NOT OVER
C***** MATRIX T
  115 DO 150 J=1,N
      DO 150 JJ=1,N
      SU = 0.
      DO 120 I=1,KTO
      RIJ = A(I,J)
      RKJ = A(I,JJ)
  120 SU = SU +  C(I) / ABS(C(I)) *RIJ * RKJ * W(I)
      I2 = KTO
      SUD = 0.
      DO 140 L=2,M1
      I1 = I2 + 1
      I2 = I1 + KT(L) - 1
      SUS = 0.
      DO 130 I=I1,I2
      RIJ = A(I,J)
      RKJ = A(I,JJ)
  130 SUS= SUS+  C(I) / ABS(C(I)) *RIJ * RKJ * W(I)
      L1 = L - 1
      SUL = SUS * AMD(L1)
  140 SUD = SUD + SUL
  150 R(J,JJ) = SUD  - SU
      GO TO 190
C***** THE FIRST ITERATION IS OVER
C*****
C***** NEW WEIGHTS GENERATED BY THE ALGORITHM
  155 CALL GP3
      IF (IT .LT. 11)  GO TO 1
      WRITE (LZ,1090)
```

```
                                   (I,W(I), I=1,KTO)
    1 CONTINUE
      SUM = 0.
      DO 160 I=1,KTO
  160 SUM = SUM + W(I) * (C(I)/ ABS(C(I)))
      DO 170 I=1,KTO
  170 W(I) = W(I)/V
C***** NEW VECTOR AND MATRIX OF ORTHOGONAL CONDITIONS
      CALL GF22
      DO 180 L=1,M
      IR = L + N1
  180 AMD(L) = AMD(L) + PIVOT(IR)
      IF (NOT .EQ. 0)  AMD(1) = -1.
      WRITE (LZ,1091)
      WRITE (LZ,1050)  (I,AMD(I), I=1,M)
      GO TO 115
C-----------------------------------------------------------------------
C***** EVALUATION OF ERROR
  190 SUMV = 0.
      DO 200 I=1,KTO
  200 SUMV = SUMV + W(I) *(C(I)/ ABS(C(I)))
      DO 220 IR=1,N
      E(IR) = 0.
      DO 210 JR=NR,MR
      J = JR - N1
  210 E(IR) = E(IR) + R(IR,JR) * AMD(J)
  220 E(IR) = R(IR,N1) - E(IR)
      E(N1) = S - SUMV
      I2 = KTO
      DO 240 L=2,M1
      I1 = I2 + 1
      I2 = I1 + KT(L) - 1
      SG = 0.
      DO 230 I=I1,I2
  230 SG = SG + W(I) * (C(I)/ ABS(C(I)))
      IR = L + N1 - 1
  240 E(IR) = 1. - SG
      IF (IT .LT. 11)  GO TO 2
      WRITE (LZ,1092)
      WRITE (LZ,1050) (I,E(I),I=1,MR)
    2 CONTINUE
C-----------------------------------------------------------------------
      WRITE (LZ,1160) IT
      IF (IT .LT. 11)  GO TO 4
      DO 250 J=1,N
  250 WRITE (LZ,1150) J,X(J)
      WRITE (LZ,1170)
      VV = V * S
      SUMT = SUM
      WRITE (LZ,1130) SUMT
      WRITE (LZ,1140) VV
    4 CONTINUE
C-----------------------------------------------------------------------
 1130 FORMAT (34X,18HOBJECTIVE FUNCTION ,1H=,F15.8)
 1140 FORMAT (34X,18HDUAL FUNCTION      ,1H=,F15.8)
```

```
1160  FORMAT (          1X,33H--------------------  ITERATION ,I4,21H------
     1-------------- )
1170  FORMAT (1X)
C-------------------------------------------------------------------------
C***** FINAL EXPRESSION OF THE NEWTON-RAPHSON MATRIX
      DO 260 IR=N1,MR
      DO 260 JR=1,N
 260  R(IR,JR) = R(JR,IR)
      DO 270 IR=N1,MR
      DO 270 JR=N1,MR
 270  R(IR,JR) = 0.
      R(N1,N1) = -SIG
C-------------------------------------------------------------------------
C***** INVERSION OF THE MATRIX R
      MM = 1
      CALL GP10
C***** THE INVERSE OF THE MATRIX R IS MULTIPLIED BY THE VECTOR
C***** E IN ORDER TO FIND THE VECTOR OF ADJUSTMENTS
      DO 280 IR=1,MR
      PIVOT(IR) = 0.
      DO 280 JR=1,MR
 280  PIVOT(IR) = PIVOT(IR) + R(IR,JR) * E(JR)
      IF (IT .LT. 11)  GO TO 3
      WRITE (LZ,1093)
      WRITE (LZ,1050) (I,PIVOT(I), I=1,MR)
   3  CONTINUE
C***** THE NEW ITERATION STARTS HERE
      IT = IT + 1
C***** THE PRIMITIVE SOLUTION IS MODIFIED AND A
C***** NEW VALUE OF V IS FOUND
      DO 290 J=1,N
      RIJ = X(J)
      PI = PIVOT(J)
      IF (PI .GT. 88.)  PI = 87.
 290  X(J) = RIJ * EXP(PI)
      PI = PIVOT(N1)
      V = V * EXP(PI)
      DO 300 IR=1,N
      DO 300 JR=N1,MR
 300  R(IR,JR) = 0.
C***** TEST OF THE SOLUTION
      CONVG = .001
      IF ( ABS(V-SIG*SUM)/V - CONVG)  320,320,310
 310  CONTINUE
C***** OPTIMIZATION SUB-LOOP ENDS HERE
         WRITE (LZ,1190)
      GO TO 330
C***** SUB-LOOP SOLUTION SATISFACTORY
 320  WRITE (LZ,1180)
      MOT = MOT + 1
      IF (MOT .LE. 6)  GO TO 6
 330  DO 340 J=1,N
 340  WRITE (LZ,1150) J,X(J)
      WRITE (LZ,1170)
```

```
    6 CONTINUE
      WRITE (LZ,113Ø) SUM
  371 L1 = L1 + 1
      DO 342 I=1,KIT
  342 C(I) = CC(I)
      CALL GF3
      ST = Ø.
      DO 345 I=61,KIT
  345 ST = ST + W(I)
      KK = 41
      TR = ST
      IF (MOT .GT. Ø)  WRITE (LZ,1Ø5Ø) (I,W(I),I=1,KIT), KK, ST
      DO 35Ø I=21,6Ø
  35Ø ST = ST + W(I)
C***** SEPARATION OF WEIGHTS INTO CATEGORIES
      MO = MOT - 3
      IF (MO .LT. Ø)  MO = 1
      L3 = Ø
      LP = L2
      L2 = Ø
      L4 = Ø
      W(61) = TR
      MTO = MOT - 1
      IF (MOT .EQ. 1)  GO TO 385
      DO 351 I=21,61
      J = I - 2Ø
      IF (K(J) .NE. Ø)  GO TO 351
      IF (W(I) .LT. 1.Ø)  GO TO 351
      KTN(J) = 1
      K(J) = 1
      STO(J) = SUR(J)
      SUR(J) = SO(J)
      L2 = L2 + 1
      LA(L2) = I
  351 CONTINUE
C***** AFTER EVERY LR OUTER-LOOP ITERATIONS RECATOGARISE TIGHT
C***** CONSTRAINTS WHICH HAVE BECOME LOOSE
      Y = Ø.85
      LR = 5
      LY = MOD(MOT,LR)
      IF (LY .NE. Ø)  GO TO 355
      DO 1352 I=21,61
      IF (W(I) .GT. 1.2)  GO TO 1353
 1352 CONTINUE
      GO TO 36Ø
 1353 Y = Ø.85
      N2 = MOT + 1
      DO 352 I=1,41
      II = I + 2Ø
      IF (K(I) .EQ. Ø)  GO TO 352
      IF (W(II) .GT. Y)  GO TO 352
      L3 = L3 + 1
      MA(L3) = II
      K(I) = Ø
  352 CONTINUE
```

```
[illegible]
      IF (MOT .NE. N2)  GO TO 36Ø
      DO 358 I=1,M1
      II = I + 2Ø
      IF (K(I) .NE. Ø)  GO TO 358
      IF (W(II) .LT. 1. )  GO TO 358
      L2 = L2 + 1
      LA(L2) = II
      SUP(I) = Ø.5 * SO(I)
      K(I) = 1
358   CONTINUE
36Ø   LP =L2
      DO 39Ø I=21,61
      II = I - 2Ø
      IF (LP .EQ. Ø)  GO TO 367
      DO 365 J=1,LP
      IF (I .EQ. LA(J)) GO TO 38Ø
365   CONTINUE
367   IF (MOT.GT.LP .AND. K(II).GE.1 .AND. K(II).LE.MC .AND. W(I).GT.1.)
     1 GO TO 368
      IF (K(II) .EQ. Ø)  GO TO 38Ø
      IF (W(I) .GT. 1. )  GO TO 38Ø
368   L2 = L2 + 1
      LA(L2) = I
38Ø   CONTINUE
      IF (MOT .NE. 2)  GO TO 383
      DO 382 I=1,41
382   STO(I) = SO(I)
      GO TO 385
383   DO 384 I=1,41
      IF (K(I).GE.MTO .AND. WEX(I).GE.1. .AND. SO(I).LT.STO(I))
     1 STO(I) = SO(I)
      IF (SO(I) .GT.STO(I))  GO TO 384
      STO(I) = SO(I)
384   CONTINUE
385   DO 386 I=1,41
386   SO(I) = SUR(I)
      LP = L2
39Ø   DO 42Ø I=21,61
      II = I - 2Ø
      IF (SUP(II) .LT. OT)  GO TO 4Ø5
      IF (L2 .EQ. Ø)  GO TO 394
      DO 392 J=1,LP
      IF (I .EQ. LA(J))  GO TO 42Ø
392   CONTINUE
394   IF(L3 .EQ. Ø)  GO TO 398
      DO 395 J=1,L3
      IF (I .EQ. MA(J))  GO TO 42Ø
395   CONTINUE
398   CONTINUE
      IF (W(I) .LT. Ø.95)  GO TO 4Ø5
      IF (W(I) .GT. Ø.99)  GO TO 41Ø
4ØØ   L2 = L2 + 1
      LA(L2) = I
      GO TO 42Ø
```

```
      K(II) = Ø
      NA(L3) = I
      GO TO 42Ø
 41Ø  CONTINUE
      K(II) = K(II) + 1
      L4 = L4 + 1
      NA(L4) = I
 42Ø  CONTINUE
      TOT = Ø.
 426  DO 43Ø I=1,L3
      II = NA(I) - 2Ø
 43Ø  TOT = TOT + SUR(II)
      OT = Ø.
C***** UPDATE SURROGATE CONSTANTS OF CONSTRAINTS WHICH ARE
C***** ALMOST TIGHT
      IF (MOT .EQ. 1)  GO TO 433
      DO 432 I=1,L2
      II = LA(I) - 2Ø
      J = LA(I)
      SR = STO(II) - SUR(II)
      WJ = W(J)
      SOR = SR/SUR(II)
      IF (SOR .GT. Ø.5)  WJ = .99
      IF (WJ .LE. Ø.9)  STO(II) = Ø.5 * STO(II)
      SR = STO(II) - SUR(II)
      IF (ABS(SR) .LE. Ø.ØØØØ1)  GO TO 432
      SR = SR/2.
      WJ = W(J)
      IF (WJ.GT.Ø.99  .AND.  WJ.LT.1.Ø1)  SR = Ø.
      IF (KIN(II).EQ.1  .AND.  SR.EQ.Ø.)  SUR(II) = STO(II)
      KIN(II) = Ø
      SUR(II) = SUR(II) + SR
      TOT = TOT - SR
 432  CONTINUE
 433  CONTINUE
      IF (TOT .LT. Ø.Ø)  TOT = Ø.Ø
C***** UPDATE SURROGATE CONSTANTS OF CONSTRAINTS WHICH ARE TIGHT
      DO 435 II=1,L4
      I = NA(II)
      J = I - 2Ø
      EX = 1.
      IF (K(J) .EQ. MOT)  EX = 8.
      IF (MOT .EQ. 1)  EX = 1.
      OTT = TOT * (W(I)/ST) * EX
      IF (W(I) .LE. 1.ØØ8)  OTT = Ø.
      SUR(J) = SUR(J) + OTT
      OT = OT + OTT
 435  CONTINUE
      OT = TOT - OT
      OT = OT/FLOAT(L3)
C***** UPDATE SURROGATE CONSTANTS OF CONSTRAINTS WHICH ARE LOOSE
      DO 44Ø I=1,L3
      II = NA(I) - 2Ø
 44Ø  SUR(II) = OT
```

```
      IF (SUR(J) .LE. 0.)  SUR(J) = 0.
  441 CONTINUE
      DO 500 I=1,41
      J= I + 20
  500 WEX(I) = W(J)
      IF (MOT .LT. 3)  GO TO 2100
      SUM = 0.
      DO 2000 I=1,41
 2000 SUM = SUM + SUR(I)
      DO 2010 I=1,41
 2010  SUR(I) = SUR(I)/SUM
 2100  CONTINUE
C***** NORMALISE SURROGATE CONSTANTS
      SUM = 0.
      DO 437 I=1,41
  437 SUM =  SUM + SUR(I)
      PRINT 501, KK, SUM
      CONVG = 0.01
C***** CHECK FOR CONVERGENCE IN SURROGATE (OUTER) LOOP
      DO 370 I=21,61
      W(I) = W(I) - 1.
      IF (W(I) .GT. CONVG)  GO TO 52
  370 CONTINUE
      SUM = 0.
      DO1437 II=1,L2
      I = LA(II) - 20
 1437 SUM = SUM + SUR(I)
      DO 438 II=1,L4
      I = NA(II) - 20
  438 SUM = SUM + SUR(I)
      OT = (1. - SUM) / FLOAT(L3)
      DO 439 II=1,L3
      I = MA(II) - 20
  439 SUR(I) = OT
      WRITE (L2,1185)
C---------------------------------------------------------------------------
 1180 FORMAT (1H0,45X,15HOPTIMAL RESULTS)
 1185 FORMAT (10(1H*), 32H CONVERGENCE CRITERIA SATISFIED , 10(1H*))
 1190 FORMAT (30X,39HNOT ENOUGH CONVERGENCE IN 20 ITERATIONS  )
C---------------------------------------------------------------------------
      KOU = KOU + 1
      IF (KOU .GE. 3)  STOP
      MOT = 2
C***** SURROGATE (OUTER) LOOP ENDS HERE
      GO TO 52
  445 STOP
      END
C***************************************************************************
      SUBROUTINE GP10
C***************************************************************************
C*****THE SUBROUTINE GP10 IS USED TO INVERT THE NEWTON - RAPHSON
C*****MATRIX OR ONE SUBMATRIX OF IT
C
      DIMENSION C(70),X(20),KI(2),W(70),A(70,20),R(22,22),PIVOT(22),
```

```
      COMMON  W,PIVOT,E,R,SUM,V,SIG,ARD,C,X,KI,A,MM,N,M,M1,KTT,KTC,MR,
     1M1,IT,MR1,MR
      DO 35Ø JR=MM,MR
 35Ø  IPVOT(JR) = Ø
      DO 44Ø IR=MM,MR
C*****INVESTIGATION OF THE PIVOT
      T = Ø.
      DO 385 JR=MM,MR
      IF (IPVOT(JR) - 1)  355,385,355
 355  DO 38Ø K=MM,MR
      IF (IPVOT(K)-1) 36Ø,38Ø,455
 36Ø  IF ( ABS(T) -  ABS(R(JR,K)))  37Ø,38Ø,38Ø
 37Ø  IROW = JR
      ICOL = K
      T = R(JR,K)
 38Ø  CONTINUE
 385  CONTINUE
      IPVOT(ICOL) = IPVOT(ICOL) + 1
C*****THE PIVOT IS LOCATED ON THE DIAGONAL
      IF (IROW - ICOL)  387,4ØØ,387
 387  DO 39Ø LL=MM,MR
      T = R(IROW,LL)
      R(IROW,LL) = R(ICOL,LL)
 39Ø  R(ICOL,LL) = T
 4ØØ  INDEX(IR,1) = IROW
      INDEX(IR,2) = ICOL
      PIVOT(IR) = R(ICOL,ICOL)
C*****THE PIVOT ROW IS DIVIDED BY THE PIVOT
      R(ICOL,ICOL) = 1.
      DO 41Ø LL=MM,MR
 41Ø  R(ICOL,LL) = R(ICOL,LL) / PIVOT(IR)
C*****ROWS WITHOUT PIVOT ARE REDUCED
      DO 44Ø KL=MM,MR
      IF (KL - ICOL)  42Ø,44Ø,42Ø
 42Ø  T = R(KL,ICOL)
      R(KL,ICOL) = Ø.
      DO 43Ø LL=MM,MR
      TT = T
 43Ø  R(KL,LL) = R(KL,LL) - R(ICOL,LL) * TT
 44Ø  CONTINUE
C*****COLUMNS ARE EXCHANGED
      DO 452 IR=MM,MR
      LL = MR - IR + MM
      IF (INDEX(LL,1) - INDEX(LL,2))  445,452,445
 445  JROW = INDEX(LL,1)
      JCOL = INDEX(LL,2)
      DO 45Ø K=MM,MR
      T = R(K,JROW)
      R(K,JROW) = R(K,JCOL)
      R(K,JCOL) = T
 45Ø  CONTINUE
 452  CONTINUE
 455  RETURN
      END
```

```
C***********************************************************************
C
C*****THE SUBROUTINE GP3 CALCULATES THE ABSOLUTE VALUE OF ALL
C*****THE TERMS OF THE MODEL
      DIMENSION C(70),X(20),KT(2),W(70),A(70,20),R(22,22),PIVOT(22),
     1AMD(1),E(70)
      COMMON  W,PIVOT,E,R,SUM,V,SIG,AMD,C,X,KT,A,MM,N,M,M1,KTI,KTO,MR,
     1N1,IT,MR1,NR
      I2 = 0
      DO 470 L=1,M1
      I1 = I2 + 1
      I2 = I1 + KT(L) - 1
      DO 470 I=I1,I2
      W(I) = 1.
      DO 460 J=1,N
      RBJ = X(J)
      IF (RBJ .EQ. 0.0)  GO TO 460
      AT = A(I,J)
      IF (AT .EQ. 0.0)  GO TO 460
      RBJ = RBJ**AT
      W(I) = W(I) * RBJ
  460 CONTINUE
  470 W(I) = W(I) *  ABS(C(I))
      RETURN
      END
C***********************************************************************
      SUBROUTINE GP22
C***********************************************************************
C
C*****THE SUBROUTINE GP22 IS USED TO FIND THE VECTOR OF ORTHOGONAL
C*****CONDITIONS FOR THE OBJECTIVE, AND THE MATRIX OF THESE CONDITIONS
C*****FOR THE CONSTRAINTS
C
      DIMENSION C(70),X(20),KT(2),W(70),A(70,20),R(22,22),PIVOT(22),
     1AMD(1),E(70)
      COMMON  W,PIVOT,E,R,SUM,V,SIG,AMD,C,X,KT,A,MM,N,M,M1,KTI,KTO,MR,
     1N1,IT,MR1,NR
      DO 480 IR=1,N
      I2 = 0
      DO 480 JR=N1,MR
      L = JR - N1 + 1
      I1 = I2 + 1
      I2 = I1 + KT(L) - 1
      J = IR
      DO 480 I=I1,I2
      RJJ = A(I,J)
  480 R(IR,JR) = R(IR,JR) + C(I) /  ABS(C(I)) * RJJ * W(I)
      RETURN
      END
```

APPENDIX C

PROGRAM FOR GROUPING OF OPERATIONS

Usage:

The program consists of a main program and five subroutines BUILD, KMEANS, SINGLE, MIN and MAX. The main program reads the data and prints the results.

Subroutines Required:

Subroutine MIN finds the minimum of a set of values. Subroutine MAX finds the maximum of a set of values. Subroutine SINGLE updates the statistics for each group. Subroutine KMEANS determines the closest group for each operation. Subroutine BUILD initializes the statistics of each group and calls KMEANS. All linkage between the main program and the subroutines is through ar uments of the CALL statements.

Description of Parameters:

K	Number of spindles
M	Number of operations
N	Number of characteristics for each operation
ITER	Number of iterationsfor convergence of the Mac Queen procedure
ENC	Increment or decrement for the divisor which adjusts the procedence values ($-1 < ENC < 1$)

W Vector of maximum values of all characteristics

WR Vector of maximum values of all characteristics for radial operations

WA Vector of maximum values of all characteristics for axial operations.

A Array of characteristics for all operations

TIM Vector of cutting times for all operations

OLD Vector of cutting lengths of all operations

MAS Vector of consisting of binary values: 0 indicates a radial operation and 1 indicates an axial operation

NAS Vector of operation numbers (corresponding to Fig.1)

DCLUS Vector of distances from group centroid

DIMENSION Requirements:

The DIMENSION statements should be modified according to the requirements of each particular problem. The parameters in parentheses included in the following statements conform to the parameter definitions above:

```
DIMENSION     WR(N), WA(N), TIM(M), OLT(M), FD(K,M),
              MCLU (K,M), A(M,N), W(N), SUM (8,N,K),
              X(N), NCLUS(M),  DCLUS (M)
DIMENSION     MA(M), MAS(M), NA(M), NAS(M), B(M,N), DX(M),
              JCLUS (M), X1(M), X2(M), X3(M), X4(M),NST(M)
```

Input Formats:

Card Type	Format	Contents
1	(7 F12.4,13X I 1, I3)	(A(I,J), J=1,N), TIM(I), OLT(I),MAS(I), MAS(I) (Values corresponding to the Jth characteristic of the Ith operation) (One card is required for the Ith operation, I = 1, M) W(J), J = 1, N WR(J), J = 1, N WA(J), J = 1, N ENC

Output:

The information in the print-out includes the group numbers, members of the group and maximum and minimum speed and feed limits for each group. The multiplication factor for the precedence values is also printed. All this information is repeated for 5 different values of multiplication factor.

Summary of User Requirements:

1. Formulate problem according to requirements explained in Section 3.3.3.
2. Specify values for K, M, N, ITER in the DATA statement.

3. Multiplification factor is given by [1/(V(6) - ENC)] so specify value of ENC so as to obtain required range for precedence values.

4. Adjust DIMENSION statements in main program and subroutines.

5. Modify FORMAT statement 2 and the associated READ statements for the particular problem being solved.

```
C-----------------------------------------------------------------------
C***** PROGRAM FOR CLUSTERING OPERATIONS ONTO THE SPINDLES AVAILABLE
C-----------------------------------------------------------------------
      DIMENSION WR(7),WA(7),TIM(10),OLT(10),FD(5,10),MCLU(5,10)
      DIMENSION A(10,7),W(7),SUM(8,7,5),X(7),NCLUS(10),DCLUS(10)
      DIMENSION MA(10),MAS(10),NA(10),NAS(10),B(10,7),DX(10)
      DIMENSION JCLUS(10),X1(10),X2(10),X3(10),X4(10),NST(10)
    1 FORMAT (7F12.4,2I9)
    2 FORMAT (9F7.3,13X,I1,I3)
    3 FORMAT ( 2X,I4, 9H CLUSTERS  )
    4 FORMAT ( 10X, 14HCLUSTER NUMBER ,I4)
    5 FORMAT (21X,10I9)
    6 FORMAT (/21X,11HFEED RADIAL , 4HMIN= ,F7.4,4X,4HMAX= ,F7.4/
     1 21X,10HFEED AXIAL ,1X, 4HMIN= ,F7.4,4X,4HMAX= , F7.4/
     2 21X, 5HSPEED ,6X, 4HMIN= ,F7.1,4X,4HMAX= ,F7.1 //)
    7 FORMAT(21X,I9,F11.1,F13.2,F10.3)
    8 FORMAT(27X,4HOPRN,11H SPEED(RPM),13H FEED(MM/REV),10H TIME(SEC),5H
     C H.P./)
    9 FORMAT(21X,39H MULTIPLICATION FACTOR FOR PRECEDENCE = ,F6.2/)
      DATA K/5/,M/10/,N/7/,ITER/3/
      READ (5,2)((A(I,J),J=1,N),TIM(I),OLT(I),MAS(I), NAS(I), I=1,M)
      READ (5,2)(W(I),I=1,N)
      READ (5,2)(WA(I),I=1,N)
      READ (5,2)(WR(I),I=1,N)
      READ (5,2) ENC
C***** STORE IN ORDER OF RADIAL FIRST AND THEN AXIAL OPERATIONS
      L = 0
      MO = 0
      DO 15 JJ=1,2
      IF (JJ .EQ. 2)  MO = 1
      LM = L
      DO 14 I=1,M
      IF (MAS(I) .NE. MO)  GO TO 14
      L = L + 1
      DO 13 J=1,N
      B(L,J) = A(I,J)
   13 CONTINUE
      MA(L) = MAS(I)
      NA(L) = NAS(I)
   14 CONTINUE
   15 CONTINUE
C***** STORE PRECEDENCE VECTOR
      DO 20 I=1,M
   20 DX(I) = B(I,6)
      WRITE (6,1)  ((B(I,J), J=1,7), MA(I), NA(I), I=1,10)
C***** NORMALISE FEATURE VECTORS
C***** NORMALISE RADIAL OPERATIONS WITH RADIAL MAXIMA
       CALL MAX(TIM,M,TIMAX)
      DO 22 J=1,7
      WJ=WR(J)
      DO 22 I=1,LM
   22 B(I,J)=B(I,J)/WJ
C*****NORMALISE AXIAL OPERATIONS WITH AXIAL MAXIMA
      LM1=LM+1
      DO 23 J=1,7
```

```
      WJ=WA(J)
      DO 23 I=LM1,10
   23 B(I,J)=B(I,J)/WJ
      DO 25 I=1,M
      OL=OLT(I)
      BI7=B(I,7)
      SPEED=OL/(TIMAX*BI7)
      IF(SPEED.GT.B(I,3)) SPEED=B(I,3)
      IF(SPEED.LT.B(I,4)) SPEED=B(I,4)
      TIM(I)=OL/(BI7*SPEED)
      B(I,5)=SPEED
   25 CONTINUE
      DO 200 NJ=1,5
C***** CHANGE WEIGHTS FOR PRECEDENCE
      DO  28 I=1,10
  28  B(I,6) = DX(I)/W(6)
      W(6) = W(6) - ENC
      DO 200 KK=1,K
      IF  (KK.EQ.1 .OR. KK.EQ.4)  GO TO 200
      MM = 1
      M = LM
      N = 7
      NN = 1
      K3 = 0
C***** THE NEXT FOUR LINES ARE USED WHEN ONLY OPTIMA ARE CONSIDERED
      DO 29 I=1,10
      DO 29 J=1,4
      A(I,J)=B(I,J)
  29  B(I,J)=0.0
  30  CALL BUILD(B,M,N,K,ITER,SUM,NCLUS,DCLUS,X,NA,NN,MM,KK)
      DO 35 I=5,7
      DO 35 K6=1,KK
      KK6 = K6 + K3
      A(KK6,I) = SUM(3,I,K6)
  35  CONTINUE
      IF (M .GT. LM)  GO TO 40
      M = 10
      K3 = KK
      MM = LM + 1
      GO TO 30
  40  CONTINUE
C***** THE NEXT THREE LINES ARE USED WHEN ONLY OPTIMA ARE CONSIDERED
      DO 50 I=1,10
      DO 50 J=1,4
  50  B(I,J)=A(I,J)
C***** STORE IN A MATRIX THE MAX/MIN VALUES OF SEED AND FEED FOR
C***** THE RADIAL AND AXIAL GROUPS TO PREPARE FOR RADIAL/AXIAL GROUPIN
      K1 = 1
      K2 = LM
      K3 = 0
  60  DO 80 KM=1,KK
      K4 = KM + K3
      L = 0
      DO 70 I=K1,K2
      IF (NCLUS(I) .NE. KM)  GO TO 70
```

```
      NST(I) = K4
      L = L + 1
      X1(L) = B(I,1)
      X2(L) = B(I,2)
      X3(L) = B(I,3)
      X4(L) = B(I,4)
   70 CONTINUE
       CALL MIN(X1,L,A(K4,1))
       CALL MIN(X3,L,A(K4,3))
       CALL MAX(X2,L,A(K4,2))
       CALL MAX(X4,L,A(K4,4))
   80 CONTINUE
      IF (K3 .EQ. KK)  GO TO 90
      K3 = KK
      K1 = LM + 1
      K2 = 10
      GO TO 60
   90 CONTINUE
C***** RENORMALISE WITH RESPECT TO OVERALL MAXIMA
      KK2=KK* 2
      DO 95 J=1,4
      WRJ=WR(J)
      WJ=W(J)
      DO 95 I=1,KK2
      IF(I.GT.KK) WRJ=WA(J)
   95 A(I,J)=A(I,J)*WRJ/WJ
      NN = 3
      N=6
  105 CONTINUE
      M = K4
      MM = 1
      DO 100 I=1,K4
  100 NAS(I) = I
C***** START WITH RADIAL/AXIAL GROUPING ON BASIS OF SPEED AND PRECEDENC
C***** THE NEXT FOUR LINES ARE USED WHEN ONLY OPTIMA ARE CONSIDERED
      DO 104 I=1,KK2
      DO 104 J=3,4
      FD(J,I)=A(I,J)
  104 A(I,J)=0.0
       CALL BUILD(A,M,N,K,ITER,SUM,JCLUS,DCLUS,X,NAS,NN,MM,KK)
C***** THE NEXT THREE LINES ARE USED WHEN ONLY OPTIMA ARE CONSIDERED
      DO 106 I=1,KK2
      DO 106 J=3,4
  106 A(I,J)=FD(J,I)
      DO 130 J=1,KK
      L = 0
      DO 120 I=1,K4
      IF (JCLUS(I) .NE. J)  GO TO 120
      DO 110 JJ=1,10
      IF (NST(JJ) .NE. I)  GO TO 110
      NCLUS(JJ) = J
  110 CONTINUE
      L = L + 1
      X1(L) = A(I,3)
      X2(L) = A(I,4)
```

```
  120   CONTINUE
         CALL MIN(X1,L,FD(J,6))
         CALL MAX(X2,L,FD(J,5))
  130   CONTINUE
        WRITE (6,3) KK
        DO 132 J=1,4
         CON=1.0E+20
        IF(J.EQ.2.OR.J.EQ.4) CON=0.0
        DO 132 I=1,KK
   132 FD(I,J)=CON
C***** ARRANGE AND PRINT OUTPUT
        DO 150 I=1,KK
        WRITE (6,4) I
        WJ=1./(W(6)+ENC)
        WRITE(6,9) WJ
        WRITE(6,8)
        L = 0
        DO 140 J=1,10
        IF (NCLUS(J) .NE. I)  GO TO 140
        L = L + 1
        MCLU(I,L)=NA(J)
        MAS(L)=J
        IF(MA(J).EQ.1) GO TO 136
        IF(B(J,1).LT.FD(I,1)) FD(I,1)=B(J,1)
        IF(B(J,2).GT.FD(I,2)) FD(I,2)=B(J,2)
        GO TO 138
   136 IF(B(J,1).LT.FD(I,3)) FD(I,3)=B(J,1)
        IF(B(J,2).GT.FD(I,4)) FD(I,4)=B(J,2)
   138 CONTINUE
  140   CONTINUE
        DO 142 JJ=1,L
        LL=MAS(JJ)
        WRITE(6,7) MCLU(I,JJ),B(LL,5),B(LL,7),TIM(LL)
   142 CONTINUE
        II=I
        FD(II,1)=FD(II,1)*W(1)
        FD(II,3)=FD(II,3)*W(1)
        FD(II,2)=FD(II,2)*W(2)
        FD(II,4)=FD(II,4)*W(2)
        FD(II,5)=FD(II,5)*W(3)
        FD(II,6)=FD(II,6)*W(4)
        WRITE(6,6) FD(I,2),FD(I,1),FD(I,4),FD(I,3),FD(I,6),FD(I,5)
  150   CONTINUE
  200   CONTINUE
        STOP
        END
C------------------------------------------------------------------------
        SUBROUTINE BUILD(A,M,N,K,ITER,SUM,NCLUS,DCLUS,X,NA,NN,MM,KK)
        DIMENSION SUM(8,7,5),A(10,7),X(7),NCLUS(10),DCLUS(10)
        DIMENSION NA(10)
        DO 10 I=1,8
        DO 10 J=NN,N
        DO 10 KU=1,K
   10   SUM(I,J,KU) = 0.
        DO 15 KKK=1,KK
```

```
      KJK = MM +KKK - 1
      DO 15 J=NN,N
   15 SUM(3,J,KKK) = A(KJK,J)
      DO 60 NC=1,ITER
      DMAX = 0.
      DO 30 KKK=1,KK
      DO 30 J=NN,N
      SUM(8,J,KKK) = SUM(2,J,KKK)
      IF (NC .EQ. 1)  SUM(8,J,KKK) = 1.
      SUM(2,J,KKK) = 0.
   30 SUM(1,J,KKK) = SUM(3,J,KKK)
      DO 50 I=MM,M
      DO 40 J=NN,N
   40 X(J) = A(I,J)
      NCLUS(I) = NC
       CALL KMEANS(N,KK,SUM,X,NCLUS(I),DCLUS(I),NN)
   50 CONTINUE
   60 CONTINUE
   70 CONTINUE
      DO 76 KI=1,KK
      IF (SUM(2,1,KI) .NE. 0.)  GO TO 76
      DO 75 I=1,8
      DO 75 J=NN,N
      SUM(I,J,KI) = 0.
   75 CONTINUE
   76 CONTINUE
  140 CONTINUE
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE KMEANS(N,K,SUM,X,JMIN,DMIN,NN)
      DIMENSION SUM(8,7,5),X(7)
      JMIN = 1
      DMIN = 1.0E+20
      DO 20 J=1,K
      XP = 1.0E-10
      DD = 0.
      DO 10 I=NN,N
      XSUM = ABS(X(I) - SUM(1,I,J))
      DD = DD + XSUM**2
      XP = XP + 1.
   10 CONTINUE
      DD = SQRT(DD/XP)
      IF (DD .GT. DMIN)  GO TO 20
      DMIN = DD
      JMIN = J
   20 CONTINUE
      XM = N
      DO 40 I=NN,N
   30 CALL SINGLE(X(I),SUM(2,I,JMIN),SUM(3,I,JMIN),SUM(4,I,JMIN),
     1SUM(5,I,JMIN),SUM(6,I,JMIN),SUM(7,I,JMIN))
      SUM(1,I,JMIN) = SUM(3,I,JMIN)
   40 CONTINUE
      RETURN
      END
```

```
C-----------------------------------------------------------------------
      SUBROUTINE SINGLE(X,COUNT,AVE,SD,XMIN,XMAX,SSG)
      IF (COUNT .NE. 0.)  GO TO 10
      AVE = 0.
      SD = 0.
      XMIN = 1.0E+20
      XMAX = - 1.0E+20
      SSQ = 0.
   10 COUNT = COUNT + 1.
      AVE = AVE + (X-AVE)/COUNT
      XAVE = ABS(X - AVE)
      IF (COUNT .NE. 1.)  SSG = SSG + COUNT * XAVE**2/(COUNT - 1.)
      SSQ = SGRT(SSQ/COUNT)
      IF (XMIN .GT. X)  XMIN = X
      IF (XMAX .LT. X)  XMAX = X
      RETURN
      END
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
      SUBROUTINE MIN(X,MM,DMIN)
      DIMENSION X(10)
      IF (MM .NE. 0)  GO TO 5
      DMIN = 0.
      RETURN
    5 CONTINUE
      DMIN = 1.0E+20
      DO 10 J=1,MM
      XJ = X(J)
      IF (XJ .GT. DMIN)  GO TO 10
      DMIN = XJ
   10 CONTINUE
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE MAX(X,MM,DMAX)
      DIMENSION X(10)
      IF (MM .NE. 0)  GO TO 5
      DMAX = 0.
      RETURN
    5 CONTINUE
      DMAX = -1.0E+20
      DO 10 J=1,MM
      XJ = X(J)
      IF (XJ .LT. DMAX)  GO TO 10
      DMAX = XJ
   10 CONTINUE
      RETURN
      END
```

A 58354

Date Slip

This book is to be returned on the date last stamped.

CD 6.72.9